

A NEW APPROACH FOR VISUALIZING THE SPATIAL DISTRIBUTION OF
POPULATION OVER TIME

By

Keith A. French

Submitted to the graduate degree program in the Department of Geography and the Graduate
Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of
Master of Arts.

Chairperson: Xingong Li

Stephen Egbert

Terry Slocum

Date Defended: June 26, 2015

The Thesis Committee for Keith A. French

certifies that this is the approved version of the following thesis:

A NEW APPROACH FOR VISUALIZING THE SPATIAL DISTRIBUTION OF
POPULATION OVER TIME

Chairperson: Xingong Li

Date approved: June 26, 2015

Abstract

A NEW APPROACH FOR VISUALIZING THE SPATIAL DISTRIBUTION OF POPULATION OVER TIME

Keywords: population, vector, color, direction, HSV

This research explores a new method of visualizing population dynamics using a static map medium. This involves the calculation of vectors that indicate the change in mean population center in a study area over a period of time. I calculate these vectors by examining the change in mean population center inside a moving window from one time period to the next. Using different sizes of these moving windows will yield results that tell different stories about the dynamics in the study area. The second component of this approach involves developing methods of displaying these vectors—collectively, a vector field—on a static map. Using the hue, saturation, and value (HSV) color model, I assign the hue component of color to the direction component of the vector and the value component of the color to the magnitude component of the vector. A significant challenge arises in assigning hues to particular directions. I explore this by conducting a study of respondents' association of hue with the cardinal and intermediate directions. Based on the results of this survey, I have developed my own system of hue-direction for display of vector fields. Finally, I assess the methodology by conducting a survey that presents the respondents with a few example maps and asks several questions to determine if the respondent is able to correctly read and interpret the maps. Complemented with more traditional visualizations of population change, this contribution should enhance visualization of population dynamics with static media.

Acknowledgements

First off, I would like to thank the members of my thesis committee. My advisor, Xingong Li, has always been patient with me and willing to help with anything and everything over the years I have been at the University of Kansas. His teachings were instrumental to the conception of this thesis; without him it would not exist. I would also like to thank Terry Slocum for the invaluable assistance with editing. I never realized just how much better of a thesis this could be until he provided his input. Finally, thanks goes out to Stephen Egbert for his support and his nearly-always open office door.

I would also like to thank a few others in the Geography Department, namely Bev Koerner for helping me out with administrative snags, Darin Grauberger, without whom I would never have had the ability to generate cool maps and graphics, and Mel Kroeger who ensured I had the appropriate computing resources available.

Finally, I would like to thank the friends and colleagues who offered support during the process. Lynnette Dornak, while sharing a windowless closet of an office with me, provided a voice of reason and also helped find respondents for my first survey. Zane Price and Antoinette Egitto were instrumental in helping get respondents for my second survey, along with providing moral support. I would also like to thank all my colleagues and coworkers who refused to allow me to forget that I had a thesis to finish. Their names may not be here, but I have not forgotten them.

Table of Contents

Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Figures	vii
List of Tables	xi
Chapter 1	1
Introduction	1
Chapter 2	10
Calculating Population Change Matrices	10
Data Collection and Pre-processing	14
Data Conversion: Vector to Raster	16
Calculating Mean Centers of Population in Moving Windows: Theory	17
Calculating Mean Centers of Population in Moving Windows: Implementation	21
Calculating Change in Mean Centers of Population: Theory	23
Calculating Change in Mean Centers of Population: Implementation	28
Chapter 3	29
Visualizing Population Change Vector Fields	29
Association of Color with Direction	31
Symbolizing the Direction Component of a Vector Field	39
Symbolizing the Magnitude Component of a Vector Field	41
Generating the Direction-Magnitude Composite Map	44
Streamlining the Process	46
Chapter 4	49
Presentation and Assessment	49
Map Presentation	50
Discussion of Results	53
Alternate Moving Window Sizes	55
Alternate Study Area	56
Assessing the Maps: Survey Development	60
Assessing the Maps: Survey Results	62

Chapter 5	70
Conclusions	70
Further Research	72
References	74
Appendices	76
Appendix A. Pre-processing Flowchart	76
Appendix B. Direction and Magnitude Calculation Flowchart	77
Appendix C. Code for Color Survey	78
Appendix D. The php Script to Store the Survey Results	82
Appendix E. Color Survey Raw Results	83
Appendix F. HSV Flowchart	84
Appendix G. <i>CIMC</i> Flowchart	85
Appendix H. <i>CIMC</i> Code	86
Appendix I. Post-processing Flowchart	89
Appendix J. The Map Survey	90
Appendix K. The Map Survey Results	93

List of Figures

- Figure 1.1. Choropleth map of the population change in Kansas from 1900 to 2000. As this map is simply meant to show the cartographic method, the actual values of increase and decrease are irrelevant, and therefore not shown.
- Figure 1.2. Example of a mean center of population. Given the population distribution shown in the figure, the green cross marks the mean center. Note that it is slightly off center to the southeast, influenced by the large mass of individuals living in that region.
- Figure 1.3. Mean centers of population (1790 – 2000) for the United States. From U.S. Bureau of the Census, Geography Division.
- Figure 1.4. Mean centers of population (1900 – 2000) for the states west of the Mississippi River. Calculated from U.S. Census data.
- Figure 1.5. Mean centers of population (1900 – 2000) for the state of Kansas. Calculated from U.S. Census data.
- Figure 1.6. Connecting the dots from one observation point to the next. The blue arrow represents the net change between the first observation (1900) and the last (2000).
- Figure 1.7. Net changes in mean center of population from 1900 to 2000 for (a) the entire United States, (b) the states east and west of the Mississippi, (c) each of the states, and (d) a customized 200km per cell grid.
- Figure 2.1. Net change vectors for custom 200km cell grid.
- Figure 2.2. Demonstration of moving windows to "bridge the gap" between two vectors. The gray cells are the highlighted cells from Figure 2.1. Each colored "window" is the same size as the original two cells.
- Figure 2.3. The Central Midwest study area.
- Figure 2.4. The 1900 and 2000 population datasets symbolized in choropleth maps. As these maps are intended to illustrate the raw census data, the data are not normalized.
- Figure 2.5. Simplifying and Integrating polygons. Panel (a) shows a portion of the original dataset. Panel (b) shows the result of the simplify polygon operation; note the reduced number of vertices but the addition of gaps. Panel (c) shows the result of the integrate operation, with reduced vertices and no gaps or overlap.
- Figure 2.6. A sample of enumeration units superimposed with a 1km cell grid. The single value for each enumeration unit must be "translated" to the grid structure.
- Figure 2.7. Two 5 x 5 neighborhoods centered on two randomly chosen pixels.
- Figure 2.8. A 5 x 5 neighborhood where all cell values are 1 (a) and a 5 x 5 neighborhood with a non-uniform distribution of values (b). The center gray cell is the focus cell.

Figure 2.9. The mean center of population for the 5 x 5 neighborhood.

Figure 2.10. A sample of the 5x5 moving windows over the entire dataset. The top row shows the first six, the middle row shows six more around the middle of the dataset, and the bottom row shows the final six. The blue cells are the focus cells.

Figure 2.11. The *popcent* function.

Figure 2.12. The *focalpopcent* function.

Figure 2.13. Mean centers of population for two years and the vector from one to the other.

Figure 2.14. Results of the *atan2* function. The AB vector shown in Figure 2.13 is included for comparison.

Figure 2.15. The data matrix coordinate system versus the Cartesian coordinate system and an example of the *atan2* function.

Figure 2.16. The *direction* and *magdiff* functions that calculate the direction and magnitude matrices.

Figure 3.1. An example of using arrows to symbolize a vector field at two different scales. The directions and magnitudes of the arrows in (a) are clearly visible, while those in (b) are not.

Figure 3.2. Hue band and hue wheel. The hue wheel (b) demonstrates how the hue bar (a) is continuous.

Figure 3.3. The online color survey interface.

Figure 3.4. The color survey results by direction.

Figure 3.5. The color survey results by hue. Each of the eight star-shaped figures represents one of the eight colors. The frequency that the color was chosen for a given direction is indicated by the length of the bar pointing in that particular direction.

Figure 3.6. The HSV hue wheel and the plurality result of the survey (the exact pattern chosen by the most respondents).

Figure 3.7. Direction of change in mean population center from 1900 – 2000 in the Central Midwest study area. The directions are mapped using the HSV hues.

Figure 3.8. Function to calculate the hue grid from the direction grid.

Figure 3.9. Linear vs logarithmic stretch of the magnitude of vectors to the value component range (0—1) in HSV.

Figure 3.10. MATLAB function to convert magnitude grid to value grid.

Figure 3.11. Magnitude of change in mean population center of the Central Midwest study area, 1900 – 2000, linear scaling (a), and logarithmic scaling (b).

Figure 3.12. The three test HSV component layers and their RGB conversions.

Figure 3.13. Linear (left) and logarithmic (right) test composites.

Figure 3.14. Composite map of population change between 1900 and 2000 with a neighborhood size of 55km, and using a linear magnitude stretch.

Figure 3.15. Composite map of population change between 1900 and 2000 with a neighborhood size of 55km, and using a logarithmic magnitude stretch.

Figure 4.1. Central Midwest Study Area with traditional (arrow) symbolization of the change in mean center vector field (1900 - 2000).

Figure 4.2. Translation of vector arrow symbology to HSV symbology. By placing the three example vectors (A, B, and C) onto the HSV hue-value construct, we arrive at the corresponding symbolization color.

Figure 4.3. Code to generate the hue-value compass rose legend.

Figure 4.4. Final Central Midwest linear scaling product with overlay and legend.

Figure 4.5. Central Midwest study area map with highlighted areas of interest.

Figure 4.6. The old Arapahoe County from 1900 extending from the Kansas border all the way west to Denver, and the new restructured Yuma County in 2000.

Figure 4.7. Linear and Logarithmic value scaling results for 100km and 200km moving windows in the Central Midwest study area.

Figure 4.8. Linear and Logarithmic maps of the change in mean centers of population from 1990 to 2000 in the Kansas City study area using 5, 10 and 20 kilometer moving windows.

Figure 4.9. Question 1 of the survey. The green bar represents the correct answer.

Figure 4.10. Question 2 of the survey. The green bar represents the correct answer.

Figure 4.11. Question 3 of the survey. The green bar represents the correct answer.

Figure 4.12. Question 4 of the survey. The green bar represents the correct answer.

Figure 4.13. Question 5 of the survey. The green bar represents the correct answer.

Figure 4.14. Question 6 of the survey. The green bar represents the correct answer.

Figure 4.15. Question 7 of the survey. The green bar represents the correct answer.

Figure 4.16. Question 8 of the survey. The green bar represents answers that were deemed correct.

Figure 4.17. Overall results of the survey. The orange portions of the bars represent the results from respondents that were categorized as not taking the survey seriously.

Figure 5.1. Initial hue-direction result map.

Figure 5.2. Change in mean center and choropleth map of the change in population from 1900 to 2000 at the county level for the Central Midwest study area. The yellow circles highlight urban growth areas.

Figure 5.3. Change in mean center and choropleth map of the change in population from 1990 to 2000 at the census block level for the Kansas City study area. The yellow circles highlight areas of significant increases of suburban population.

List of Tables

Table 3.1. Opponent process color choices and their respective RGB values.

Table 3.2. Color association concepts (least abstract to most abstract).

Chapter 1

Introduction

Describing dynamic phenomena with static maps has always been a cartographic challenge. Although recent technological advances have increased the availability of dynamic data exploration tools, a fair amount of everyday media are still presented in static format. The past ten years have seen significant declines in the circulation of many paper media, but there are still upward of 40 million newspapers circulated each weekday in the United States (Newspaper Association of America, 2015). News magazines, while taking a similar hit, are still circulating in the millions (Pew Research Center, 2015). The migration of textbooks to electronic versions is an ongoing process, but one recent study shows that 60 – 80% of students still use a physical textbook in their studies (Greenfield, 2013). Even with the prospect of a complete extinction of paper media, static digital maps will likely continue to endure, as static maps are still ubiquitous on the World Wide Web. As such, it is important to continue developing methods of communicating with them. This thesis explores a new methodology of communicating a dynamic data pattern using a static medium.

While the techniques described herein are applicable to a number of possible types of dynamic data, this thesis focuses on mapping trends in population over time. I have chosen to examine population change for two reasons. First and foremost, population dynamics is a topic of some interest to me. Where people choose to inhabit the earth, why they choose to do so, and why they eventually change the locations they choose to inhabit are important questions for all the social sciences. Secondly, documentation of population—at least in the United States—is fairly comprehensive, spatially and temporally, and the data are readily available.

There are already a number of established ways of representing changing population on a static map, depending on what one is trying to convey to the map reader. Choropleth maps are a common technique. These maps use color or a shade of gray in an enumeration unit (such as a county) to express the value of that unit (Slocum, McMaster, Kessler, & Howard, 2009, p. 85). The choropleth map can be used to indicate change by using a carefully devised color scheme (Figure 1.1), but in this case the map reader cannot tell anything other than the rough magnitude of gain or loss in each enumeration unit.

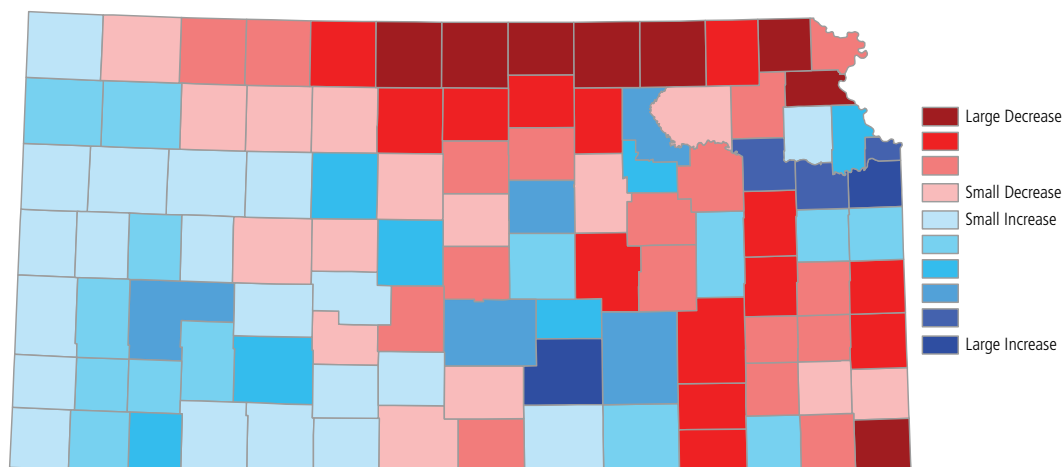


Figure 1.1. Choropleth map of the population change in Kansas from 1900 to 2000. As this map is simply meant to show the cartographic method, the actual values of increase and decrease are irrelevant, and therefore not shown.

One effort to overcome this limitation was Tobler's (1987) use of computers to create population migration maps. These maps used arrows to map migration origins and destinations based on "from-to" tables with arrows of varying sizes based on the number of migrants. While these maps were revolutionary in terms of thematic maps showing movement, they required enormous migration datasets with information on starting and ending locations. In many cases, due to the sheer amount of data, the information able to be shown on any one map was quite limited.

Another interesting technique of mapping population change employs Balzer and Deussen's Voronoi treemap (2005), which represents immigration data using variably sized and colored polygons (Aisch & Gebeloff, 2014). The polygon size represents the magnitude (relative number of immigrants) and the color represents their area of origin. The map is interactive in that the user can obtain information on any specific polygon by hovering over it; however, the map in its static form stands on its own. The drawbacks to this particular mapping technique are that specific data on the origin of individuals are required for each enumeration unit, and the computations for generating the Voronoi polygons can be quite complex.

My goal is to give the map reader an indication of population trends by calculating the change in mean centers of population. The mean center of population is the single point that indicates the gravity center of the population in a given region (Slocum, McMaster, Kessler, & Howard, 2009, p. 51). For example, if two equally populated cities are twenty miles apart, and the population distribution outside these cities is uniformly distributed, the mean center of population would lie directly between them at a point equidistant from each city. Figure 1.2 demonstrates the mean center of population of a sample population distribution.

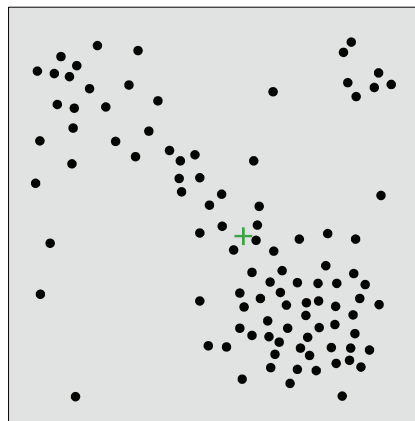


Figure 1.2. Example of a mean center of population. Given the population distribution shown in the figure, the green cross marks the mean center. Note that it is slightly off center to the southeast, influenced by the large mass of individuals living in that region.

Maps showing population trends based on the mean center of population are nothing new. The Census Bureau has published on its website <http://www.census.gov/> (United States Bureau of the Census, 2007) a map showing the general population trend of the contiguous 48 states from 1790 to 2000. The map features a series of points, each representing the geographical mean center of the entire population of the United States at ten-year intervals (Figure 1.3). An updated version of this map showing the latest geographic mean center of population is released for each census to much fanfare. Residents of the area determined to be the new geographic center will typically bank on the potential of tourism and free press.

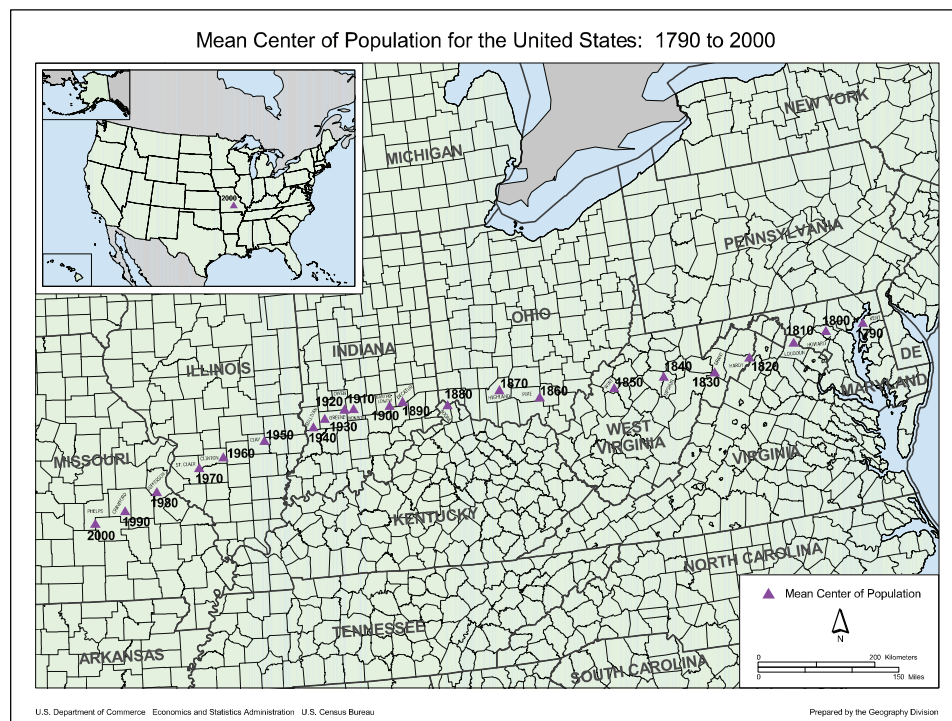


Figure 1.3. Mean centers of population (1790 – 2000) for the United States. From U.S. Bureau of the Census, Geography Division.

While this map provides an engaging visual depiction of the change in population, Barmore (1993) makes the important point that the calculation of mean centers of population is heavily affected by the choice of the map projection. The distortion in one projection will differ

from that in another; and the distortion in a single projection will differ from one part of the map to another. As this thesis is focused on the general technique and the methods of visualization, these concerns are outside its scope. It is merely important to understand the general idea behind the mean center of population changes and that the series of points represents the general trend of the change in the population center for the United States over time.

The scope of the map in Figure 1.3 is the entire United States, but if we were to change the study area used in these calculations, the resultant mean center computations would change as well. For example, if we elected to repeat the study omitting the states east of the Mississippi River in the geographical mean center of population calculation, we would end up with a considerably different map (Figure 1.4). Note that only calculations for the years 1900 – 2000 were made, due to the sparseness of overall data density in the study area before 1900.

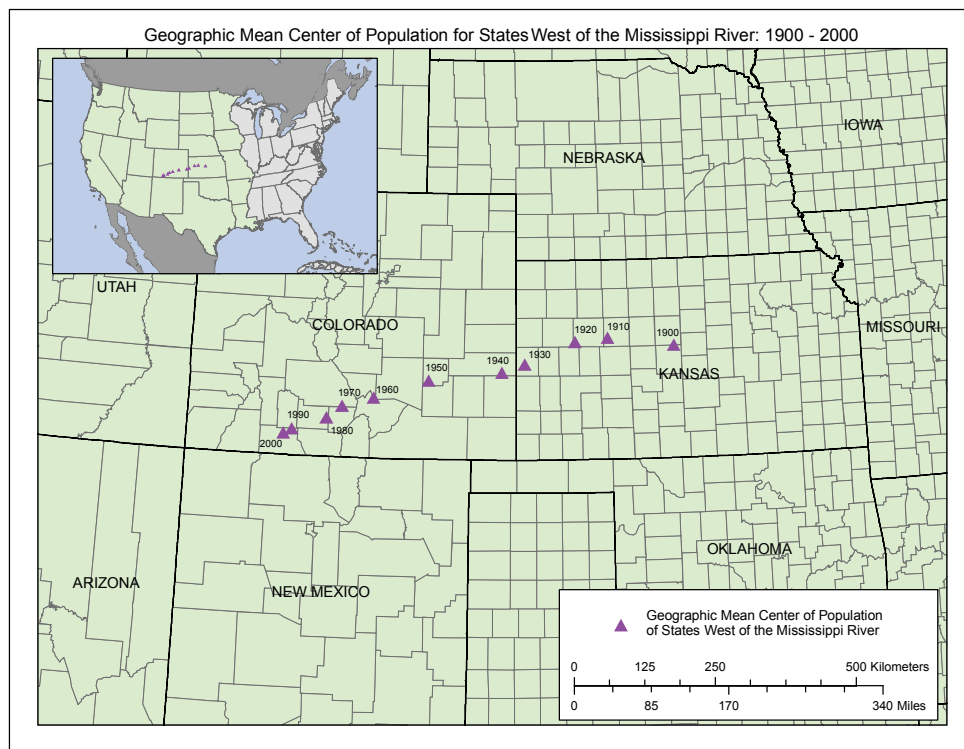


Figure 1.4. Mean centers of population (1900 – 2000) for the states west of the Mississippi River. Calculated from U.S. Census data.

Further reduction of the study area to the state of Kansas would result in an even more localized trend (Figure 1.5). In examining Figure 1.5, it is interesting to note that unlike in the first two cases, the directions from some mean center points to subsequent ones radically change in direction. From 1900 to 1930 the mean centers of population moved generally in a southwesterly direction, but this shifted briefly to the east in 1940, resuming the southwesterly direction in 1950 and then finally proceeding eastward again from 1960 to 2000.

By carefully examining the points and their labels you can get a picture of how the mean centers moved over time. These changes can be easily illustrated simply by “connecting the dots” as shown in Figure 1.6.

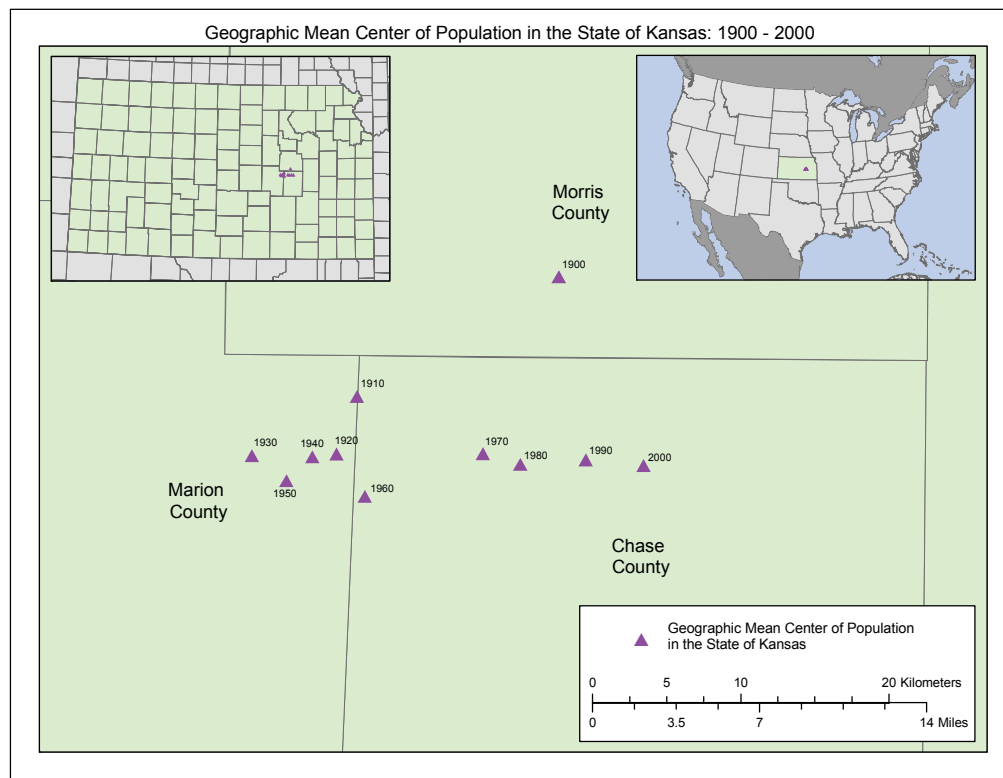


Figure 1.5. Mean centers of population (1900 – 2000) for the state of Kansas. Calculated from U.S. Census data.

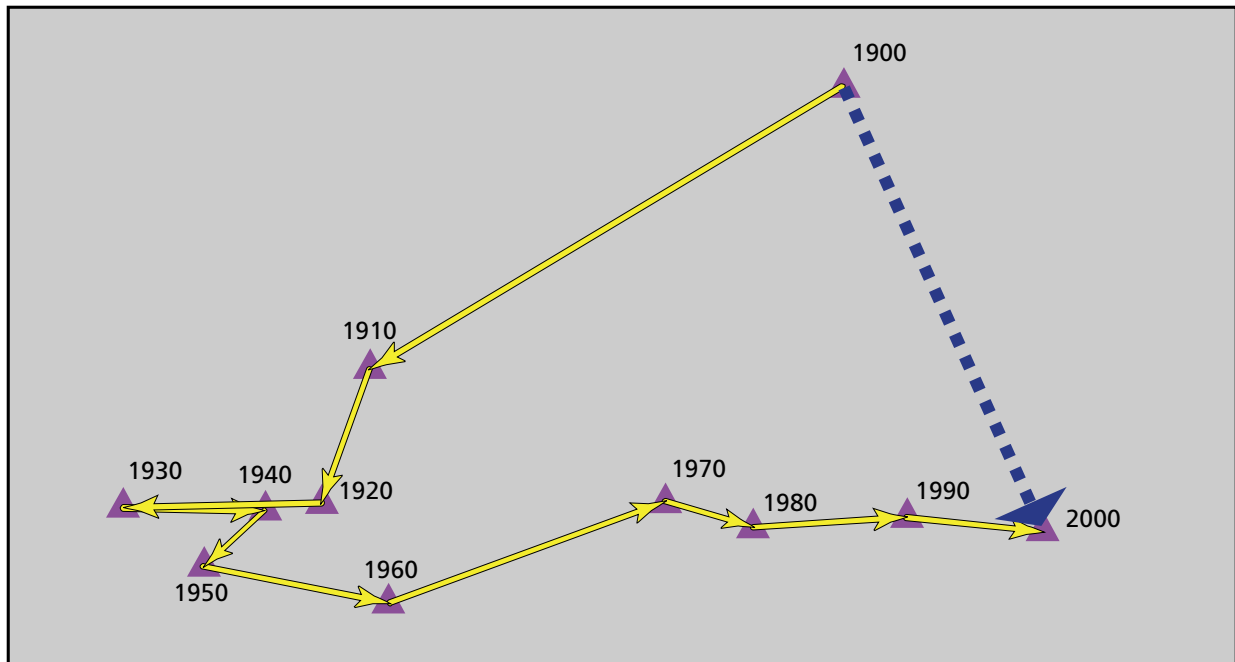


Figure 1.6. Connecting the dots from one observation point to the next. The blue arrow represents the net change between the first observation (1900) and the last (2000).

Each of the yellow arrows in Figure 1.6 shows the direction and distance of the change in mean center of population from one observation to the next (at ten-year intervals). The blue arrow summarizes the change over the entire period of time by indicating the net change from the first observation (1900) to the last (2000). It is this “net change” arrow that I will be focusing on, as it tells the story of the entire series of points in a concise manner. This will be important when we begin to consider numerous study areas over large geographic areas.

Figure 1.7 (a-c) shows these “net change” arrows for each of the previous examples for the change from 1900 to 2000. In the case of Figure 1.3, we simply draw the arrow from the 1900 point to the 2000 point (Figure 1.7a). Figure 1.7b shows the net change arrow for the states west of the Mississippi and for the states east of the Mississippi. Figure 1.7c shows the net changes for each state. Conceivably, we could take this concept down to the county level, then

the tract level, and so on until the study areas are as small as the smallest enumeration unit in which the original data are available, in this case, the census block.

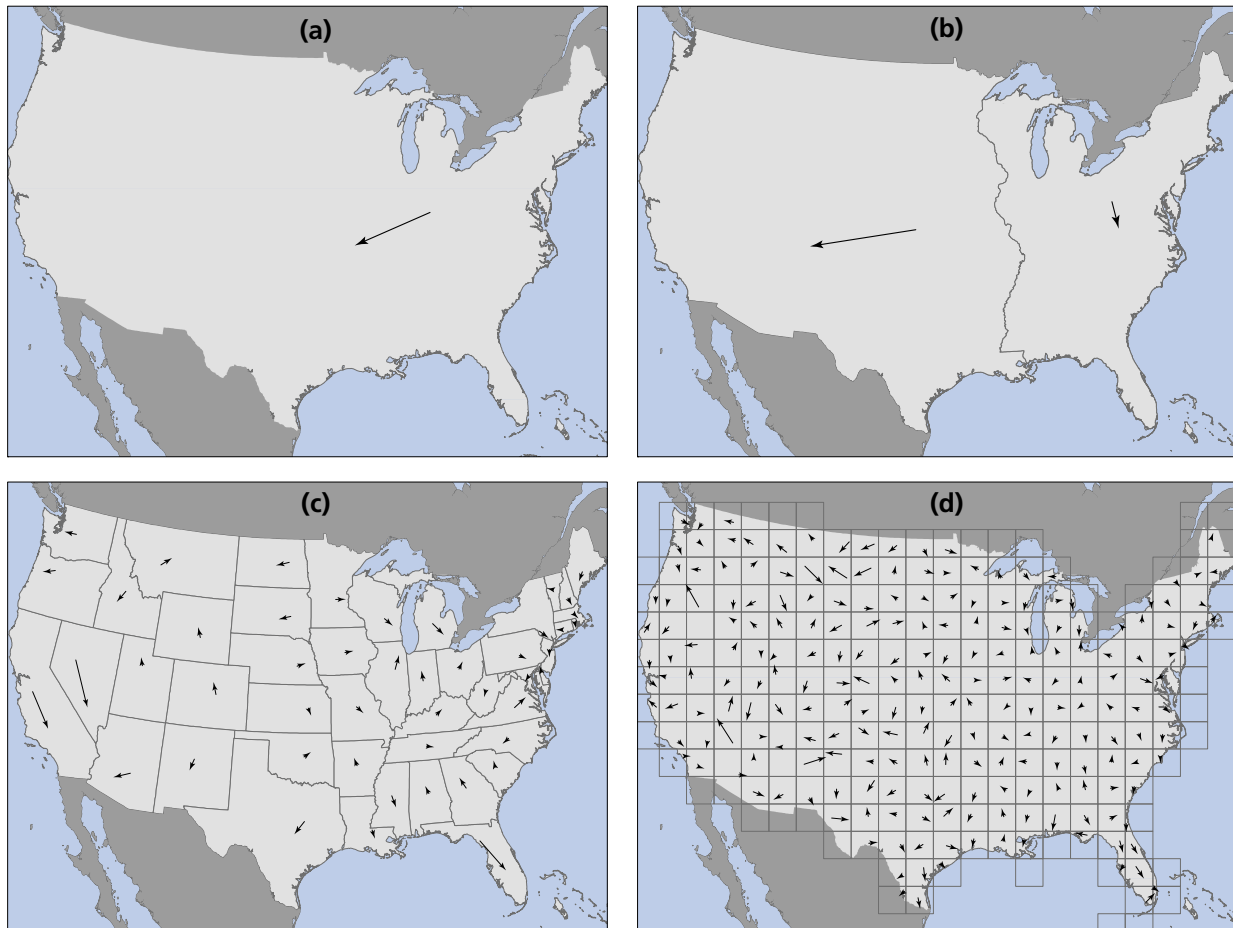


Figure 1.7. Net changes in mean center of population from 1900 to 2000 for (a) the entire United States, (b) the states east and west of the Mississippi, (c) each of the states, and (d) a customized 200km per cell grid.

We are not limited to using already defined areas such as states, counties, or tracts within which to calculate these net change arrows. We can, in fact, define our own areas in any fashion we desire. We could simply construct a grid over an area and calculate the net change arrows for each square in the grid, as shown in Figure 1.7d. In this way, we end up with a structured, continuous data matrix known as a *vector field* which I will discuss in more detail in Chapter 2.

The objective of this research is to develop and assess a method of visualizing spatial dynamics using these vector fields in a static map medium. In chapter 2, I discuss the calculations needed to generate a vector field representing the change in centers of mean population over a geographic area. In Chapter 3, I discuss the symbolization of the vector field in such a way as to represent the “net change” arrows (both magnitude and direction) without loss of spatial resolution. Finally, in Chapter 4, I assess the value of the finished maps by determining if the map reader can understand the information on the map. Maps must be legible in order to be useful, so I employ human subjects as map readers to determine their value.

Chapter 2

Calculating Population Change Matrices

At the end of Chapter 1, I introduced the notion of calculating the change in mean centers of population in a two-dimensional array of grid squares, or cells. For the sake of example, I constructed an array of cells—each 200km by 200km in size—covering the lower 48 states, as shown in Figure 2.1.

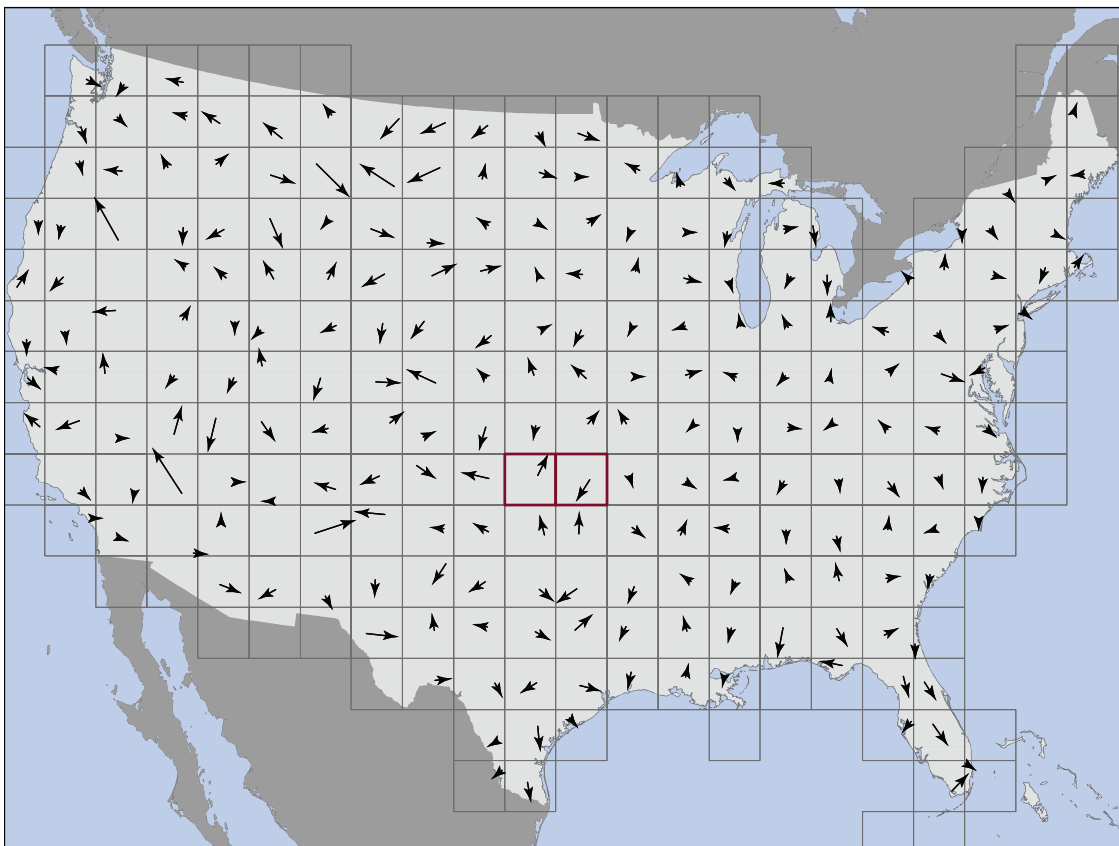


Figure 2.1. Net change vectors for custom 200km cell grid.

This array, or matrix, consists of a number of cells, each with a net change arrow, or vector, representing the change in the mean center of population from 1900 to 2000 inside each of the respective cells. This matrix of vectors is collectively known as a vector field.

Vector fields are raster data structures where the traditional scalar value of the raster cell is, instead, a vector value with both magnitude and directional components (Li & Hodgson, 2004). Each of the vectors requires two values to define it: its magnitude (the length of the arrow) and its direction. This chapter will discuss the methods of calculating these vector fields from raw census data available directly from the United States Bureau of the Census.

The selection of the 200km cell size as shown in Figure 2.1 was largely arbitrary, essentially chosen to generate a graphic where the arrows would be easily visible. With cells of this size, we are only calculating a few hundred different vectors to represent the population dynamics of the entire continental United States. This leaves enormous gaps between the arrows in the matrix. How do we bridge these gaps?

One obvious method of bridging the gaps is to simply make the cell size smaller. By making smaller cells, we make many more of them, increasing the number of the arrows while decreasing the space between them. However, by doing this we are also calculating mean centers of much smaller population distributions. If the cell sizes become smaller than the smallest aggregation of the raw census data, there will be no changes in mean centers of population, because the distribution inside the smallest units is not known, and must be assumed to be uniform.

The second method of bridging the gaps is to employ “moving windows”. In Figure 2.1, each cell occupies a 200km by 200km area, but what about other 200 km by 200 km areas on the map that do not fall precisely onto one of the cells? For example, within the area of the two red cells in Figure 2.1, there lie any given number of additional 200km by 200km cells, depending on how far we space them apart. Figure 2.2 demonstrates nine additional cells that might occur within the two existing cells and their respective vectors indicating the change in mean center of

population within those cells. By considering the vectors from all these cells, we've "bridged the gap", between the vectors of the two original cells, at least in one dimension, from west to east. Each of the different colored cells is the same size, but in a different location. If we employ moving windows in both directions (west to east, and north to south) we could bridge all the gaps in Figure 2.1.

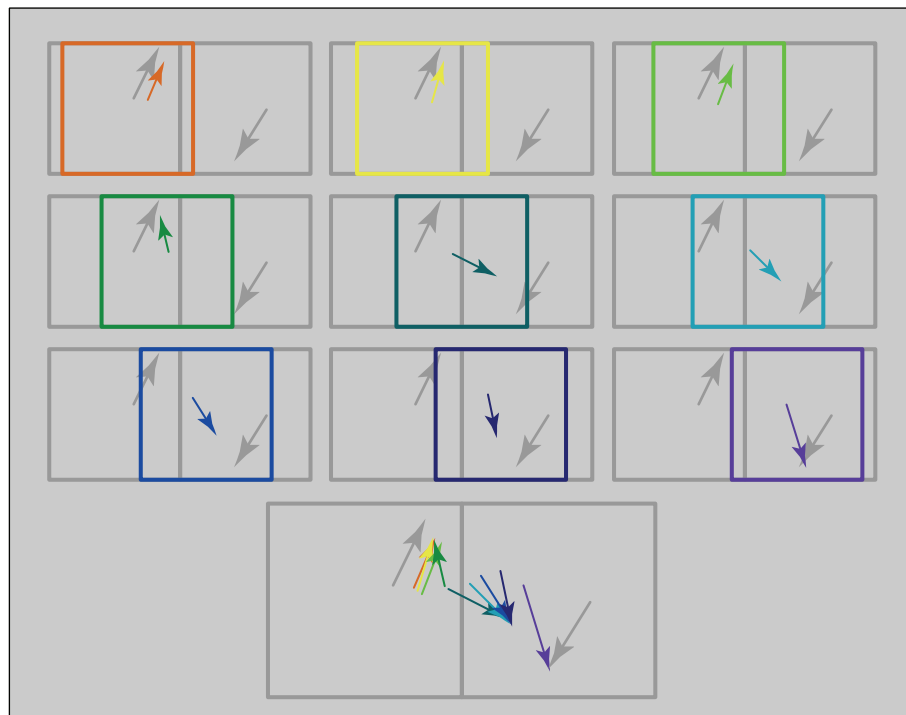


Figure 2.2. Demonstration of moving windows to "bridge the gap" between two vectors. The gray cells are the highlighted cells from Figure 2.1. Each colored "window" is the same size as the original two cells.

Tools designed specifically to perform the analysis described here do not as yet exist; however, the framework to develop such tools is available in many different geographic information system (GIS) packages. Packages such as ESRI's ArcGIS allow users to utilize low-level geoprocessing capabilities to generate their own higher-level tools that will suit their individual needs through the writing of scripts. Furthermore, since we are working with raster datasets, we are dealing with what are essentially matrices, which can be handled with

specialized matrix manipulation software such as MATLAB. MATLAB (derived from MATrix LABoratory) provides a number of functions and tools—including a proprietary programming language—to perform calculations and create reports such as graphs and tables.

My initial plan was to calculate population change vectors in moving windows covering the 48 contiguous states over the entire census period 1790 - 2000, but concerns about lengthy processing times working with enormous datasets caused me to cut the study area down. Instead, I chose to focus on the Central Midwestern United States, from the southwestern corner of Kansas to the northeastern corner of Iowa, covering most of Nebraska and Missouri as well (Figure 2.3). I chose this region because of my familiarity with it and because the counties in this area are of relatively similar size, which will be important later. Because census data are not complete in this region for the entire 1790 – 2000 period, the study period was reduced to the 1900 – 2000 timeframe.

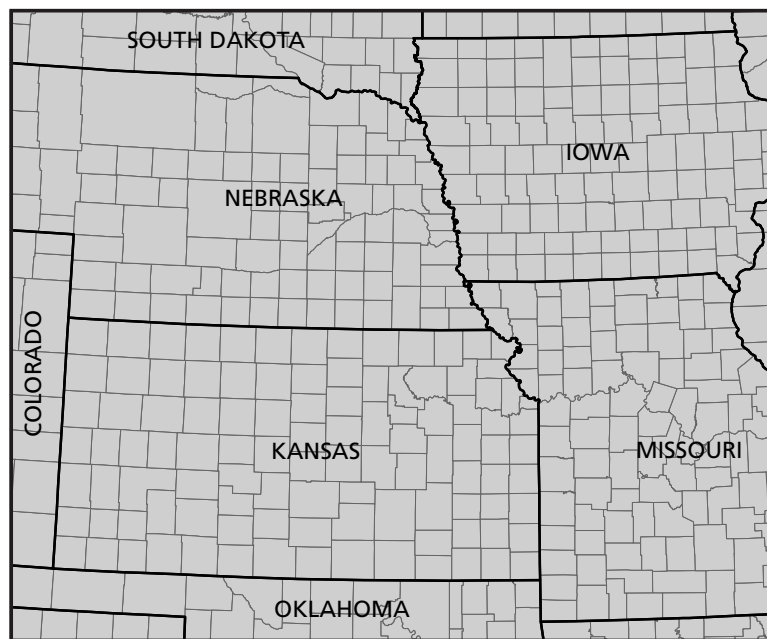


Figure 2.3. The Central Midwest study area.

Data Collection and Pre-processing

I acquired the population datasets from the National Historical Geographic Information System (NHGIS) at <http://www.nhgis.org/> (Minnesota Population Center, 2011). These datasets come in the form of high-resolution shapefiles accompanied with comma separated value (csv) files containing the actual demographic data. Additional formats are available, and a wealth of demographic data beyond mere population are also available. For this project, I acquired only total population at the county level for the years 1900 and 2000, and the associated shapefile datasets.

Upon completing the download of the shapefiles and the csv tables, the first step was to join the data tables to the shapefiles. The NHGIS datasets are very well prepared for this purpose. Each enumeration unit in the shapefiles has a unique GISJOIN identification code that corresponds with the data for that particular enumeration unit in the tables. The join was accomplished using the ESRI ArcMap *join* function. I then created choropleth maps of the study area for 1900 and 2000 so I could get an idea of what the general population distribution looks like, as shown in Figure 2.4.

It became apparent very quickly that working with the raw NHGIS shapefile datasets was going to be problematic, as these are highly detailed vector datasets with extremely high vertex counts (Figure 2.5a). Even the most basic GIS functions took enormous amounts of time; indeed, just the time required for the ArcMap software to render the datasets on the computer screen took a fair amount of time. To remedy this, I ran each of the datasets through the ESRI ArcInfo *simplify polygon* tool with a 100 meter tolerance to reduce the number of vertices down to manageable levels. Unfortunately, this process also introduced gaps and overlaps between neighboring polygons as shown in Figure 2.5b.

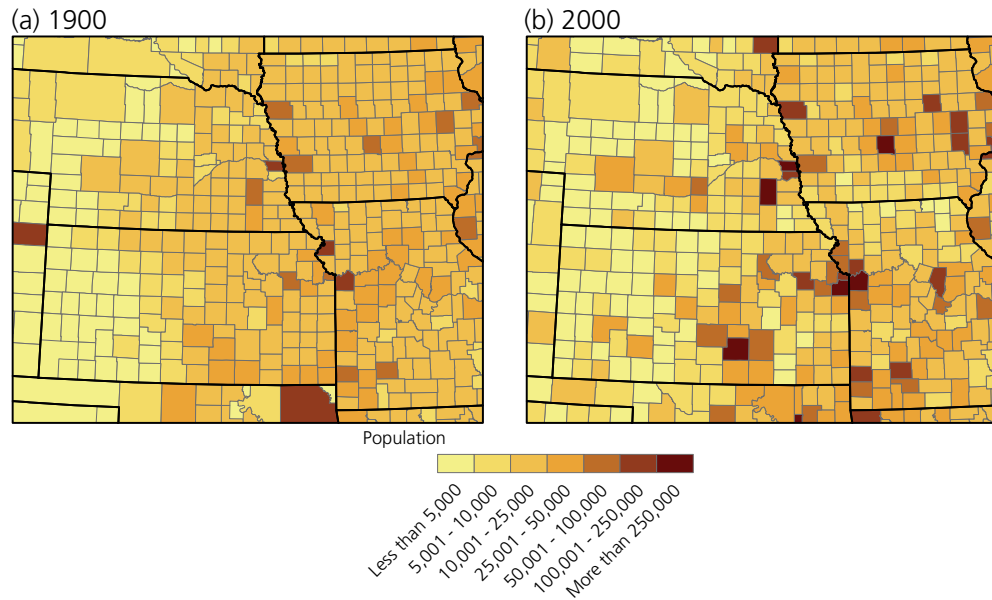


Figure 2.4. The 1900 and 2000 population datasets symbolized in choropleth maps. As these maps are intended to illustrate the raw census data, the data are not normalized.

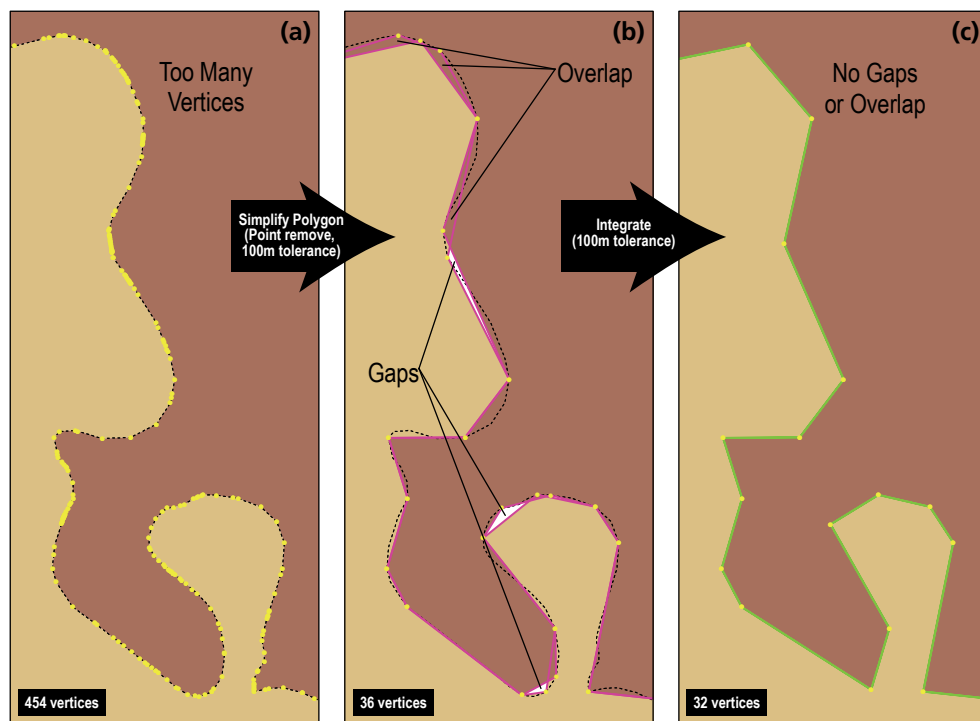


Figure 2.5. Simplifying and Integrating polygons. Panel (a) shows a portion of the original dataset. Panel (b) shows the result of the *simplify polygon* operation; note the reduced number of vertices but the addition of gaps. Panel (c) shows the result of the *integrate* operation, with reduced vertices and no gaps or overlap.

I remedied this using the ESRI ArcInfo *integrate* tool, also at a 100 meter tolerance, to close those gaps and eliminate overlapping (Figure 2.5c). The steps outlined here are shown in a flowchart form in Appendix A.

Data Conversion: Vector to Raster

As mentioned earlier, vector fields are raster data structures, and to calculate these, we must have the population data in a raster format as well. The census data are aggregated based on various enumeration units, from the smallest census block to the block group, tract, county, state and finally the national level. Since these enumeration units vary in size and shape, representing these data in a raster or grid format requires a conversion process. Figure 2.6 illustrates a grid of 1 square kilometer cells superimposed on county enumeration units.

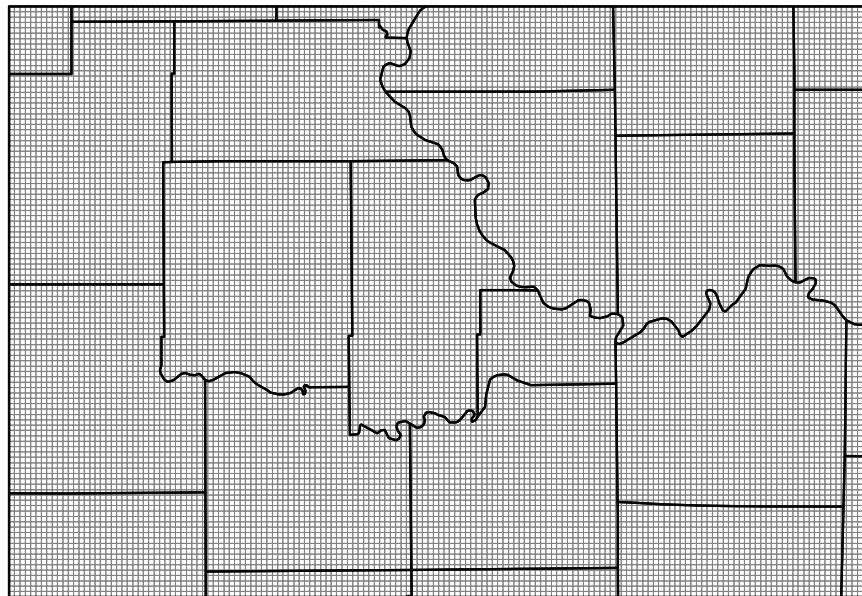


Figure 2.6. A sample of enumeration units superimposed with a 1km cell grid. The single value for each enumeration unit must be “translated” to the grid structure.

I began by calculating the area for each enumeration unit by adding a new field in each of the shapefiles called “Area” and using the ArcMap *calculate geometry* tool to calculate the area

in square kilometers. Then I added a field called “Density” and calculated the population density for each enumeration unit by dividing the population by the area, netting a value representing persons per square kilometer.

In preparation for the conversion to the raster format, I clipped the NHGIS datasets to the study area so that the output of the conversion would cover only that geographic area. I then executed the ArcMap *feature to raster* tool to make the final conversion, the result being a raster dataset where each cell’s value is that cell’s estimated population.

The raster grid over the Central Midwest study area ended up with a dimension of 1033 x 848 (1 sq. km cells), which seemed like a reasonable size to handle without running into computer memory problems or prohibitive processing times. However, the grid needed to be “ported” to a format that the MATLAB software could work with. To handle this, I converted the raster grid to ASCII raster format using the ESRI *raster to ASCII* tool, and then converted that to a MATLAB format file (.mat) using the MATLAB *grid2mat* function. The intermediate conversion to ASCII raster is necessary because the *grid2mat* function does not support the proprietary ESRI raster format as an input. I performed all of the above procedures on both the 1900 and 2000 census data, as both datasets are required to be in the same format to perform the mean center change calculations.

Calculating Mean Centers of Population in Moving Windows: Theory

The input needed to calculate the mean center of population is a raster dataset where each cell’s value is the population in the cell. Utilizing cartographic modeling techniques (Tomlin, 1990), neighborhood statistical analysis can be performed on each cell in the input matrix. At each of these cells—hereinafter referred to as focus cells—a neighborhood is delineated based on the geographic location of the focus cell and the user-chosen neighborhood configuration. This

neighborhood can be as small and as simple as the immediately adjacent raster cells, or it can also be very large, and either square or circular in shape.

Figure 2.7 shows two square 5 x 5 neighborhoods (the size chosen arbitrarily, for the sake of example) centered on two different cells in a fictitious study area. These neighborhoods are what will become our “moving windows”.

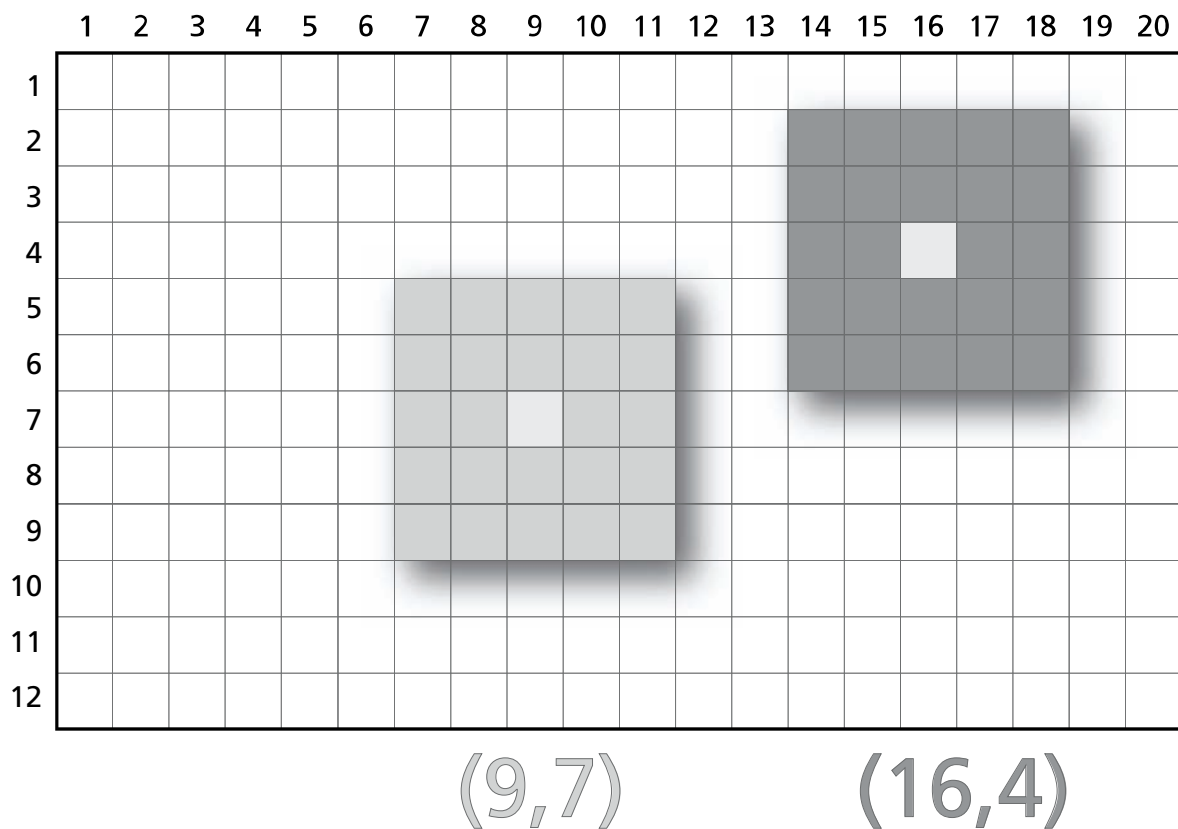


Figure 2.7. Two 5 x 5 neighborhoods centered on two randomly chosen pixels.

Once a neighborhood is established, the population weighted center of the neighborhood is calculated. For example, if we have the 5 x 5 neighborhood shown in Figure 2.8a (our focus cell being the center cell), we have a neighborhood where the population in each cell is 1, and so the population is evenly distributed over the entire neighborhood. The center of population will

therefore simply be the geographic center, which in this case is the center cell. It also happens that this cell is our focus cell.

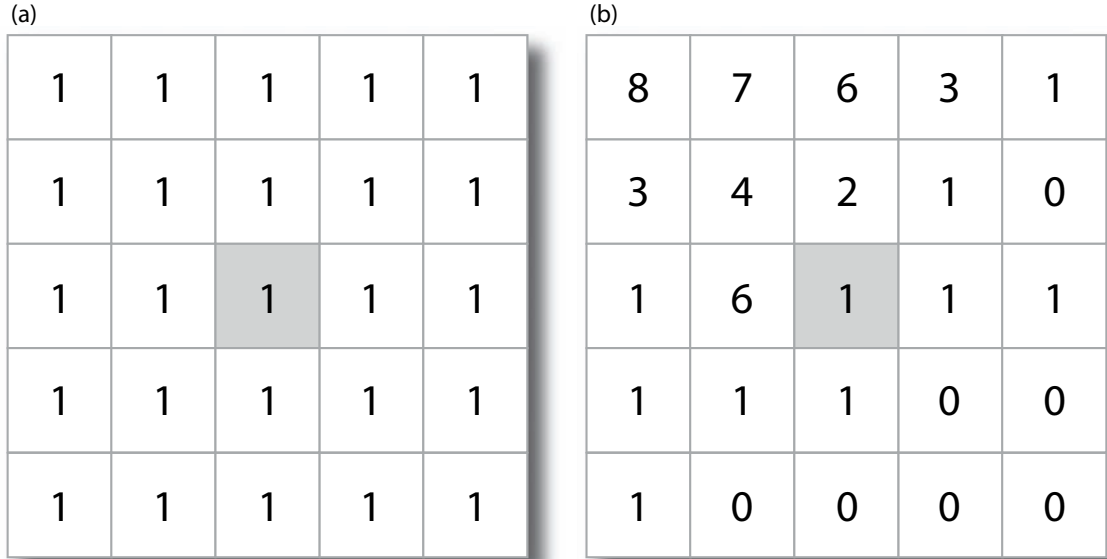


Figure 2.8. A 5 x 5 neighborhood where all cell values are 1 (a) and a 5 x 5 neighborhood with a non-uniform distribution of values (b). The center gray cell is the focus cell.

If, however we have the neighborhood shown in Figure 2.8b, we notice that there is a higher population in the northwest than in other parts of the neighborhood. Thus, we should expect our population center to be located somewhere in the northwest area. We can use the following formulas for calculating the weighted geographic mean center to see if our expectations are accurate:

$$x_c = \frac{\sum w_i x_i}{\sum w_i} \quad y_c = \frac{\sum w_i y_i}{\sum w_i}$$

where x_c and y_c will be the x and y location of the population weighted center, w_i is the weight (or population) at cell i , and x_i and y_i are the x and y locations of cell i . For x_c we calculate the numerator (going from the top left cell along each row):

$$\begin{array}{r}
(8 * 1) + (7 * 2) + (6 * 3) + (3 * 4) + (1 * 5) = 57 \\
(3 * 1) + (4 * 2) + (2 * 3) + (1 * 4) + (0 * 5) = 21 \\
(1 * 1) + (6 * 2) + (1 * 3) + (1 * 4) + (1 * 5) = 25 \\
(1 * 1) + (1 * 2) + (1 * 3) + (0 * 4) + (0 * 5) = 6 \\
(1 * 1) + (0 * 2) + (0 * 3) + (0 * 4) + (0 * 5) = 1 \\
\hline
110
\end{array}$$

For the denominator, we have:

$$\begin{array}{r}
8 + 7 + 6 + 3 + 1 = 25 \\
3 + 4 + 2 + 1 + 0 = 10 \\
1 + 6 + 1 + 1 + 1 = 10 \\
1 + 1 + 1 + 0 + 0 = 3 \\
1 + 0 + 0 + 0 + 0 = 1 \\
\hline
49
\end{array}$$

Dividing 110 by 49 yields a result of 2.2449 for x_c . We perform the same calculation for y_c and end up with a value of 1.8776. As shown in Figure 2.9, if we place a mark on the grid in roughly that x and y location, we can indeed confirm that the larger population in the northwest has had an effect on the location of the population center. The output value for the focus cell in this case will be the point location (2.2449, 1.8776).

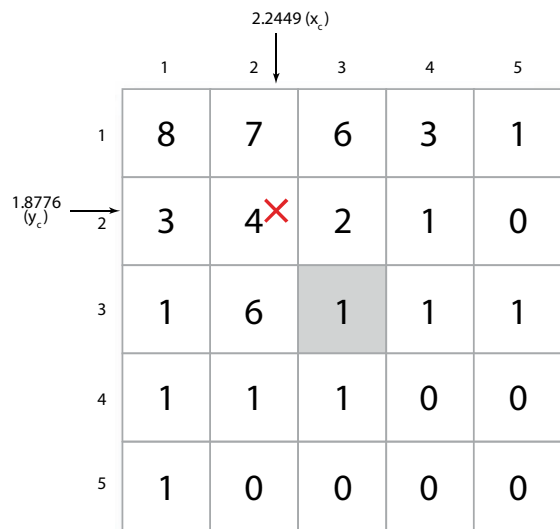


Figure 2.9. The mean center of population for the 5 x 5 neighborhood.

The procedure for the 5 x 5 neighborhood is repeated for each focus cell in the entire dataset starting at the upper left focus cell and proceeding left to right, row by row to the lower right cell as shown in Figure 2.10. Note that the neighborhoods along the periphery are not whole; only the cells inside the boundaries of the study area are included in the calculations. The x and y locations for the mean center are stored in each of the focus cells (the blue cells) in the output grid. The final product is a matrix where each cell's value is an (x, y) point defining the geographic location of its neighborhood's population center. In practice, using only standard data types the output will consist of two matrices—one for the x component and one for the y component.

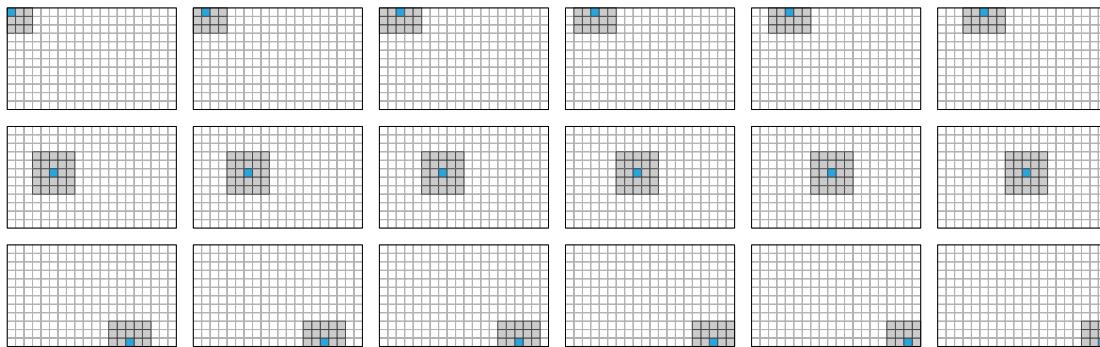


Figure 2.10. A sample of the 5x5 moving windows over the entire dataset. The top row shows the first six, the middle row shows six more around the middle of the dataset, and the bottom row shows the final six. The blue cells are the focus cells.

Calculating Mean Centers of Population in Moving Windows: Implementation

I wrote MATLAB “m-file” functions to handle all the calculations discussed in the previous section. The *popcent* function in Figure 2.11 simply returns two scalar values, x_c and y_c , representing the population weighted center of the input grid. If we were to call the function as such:

```
[x, y] = popcent(Figure2.9_Grid)
```


where “Figure2.9_Grid” is the example in Figure 2.9, the output x and y values would be 2.2449, and 1.8776 respectively, as calculated in the previous section. It is important to note that I do not use this function directly, rather it is called by the *focalpopcent* function to process a subset (neighborhood) of a much larger input grid.

```
function [x, y] = popcent(popgrid)

% USAGE: [x, y] = popcent(popgrid) where popgrid is a matrix with population
% density values (or any value you wish to act as a weight) and [x, y] are
% the x and y values of the resultant population center relative to the
% input grid with the origin at the upper left.

rows = size(popgrid, 1);
cols = size(popgrid, 2);

wsum = sum(popgrid(:));

xsum = 0; ysum = 0;
for i = 1:rows
    for j = 1:cols
        xsum = xsum + popgrid(i,j).*j;
        ysum = ysum + popgrid(i,j).*i;
    end
end

x = xsum / wsum;
y = ysum / wsum;

end
```

Figure 2.11. The *popcent* function.

The *focalpopcent* function (Figure 2.12) accepts as input the entire population grid and an integer defining the size of the desired neighborhood.

```
function [xgrid, ygrid] = focalpopcent(popgrid, window_size)

% USAGE [xgrid, ygrid] focalpopcent(popgrid, window_size) where popgrid is
% the input grid and window_size is an positive integer representing the size of
% the focal region. Code will produce warnings if window_size is an even
% integer but will still run. Outputs two grids, one for the x location and one
% for the y location.

rows = size(popgrid, 1);
cols = size(popgrid, 2);

buffer = (window_size - 1) / 2;
for i = 1:rows
    for j = 1:cols

        % determine neighborhood region
        xstart = j - buffer;
        if (xstart < 1)
            xstart = 1;
        end
        xend = j + buffer;
        if (xend > cols)
            xend = cols;
        end
        ystart = i - buffer;
        if (ystart < 1)
            ystart = 1;
        end
        yend = i + buffer;
        if (yend > rows)
            yend = rows;
        end
        nbhood = popgrid(ystart:yend, xstart:xend);
        [x, y] = popcent(nbhood);

        xgrid(i,j) = xstart - 1 + x;
        ygrid(i,j) = ystart - 1 + y;
    end
end

end
```

Figure 2.12. The *focalpopcent* function.

This function progresses through the input matrix and calls the *popcent* function for a subset of the matrix based on the input value for the neighborhood size. Much like the *popcent* function, the *focalpopcent* function returns x and y coordinates, but it does so for every neighborhood in the input data matrix. The output, therefore, is a pair of matrices, one for the x coordinates and one for the y coordinates.

The decision in determining the size of the neighborhoods themselves is largely based on the average size of the input enumeration units. As mentioned earlier, the size of the neighborhood is limited only by the spatial resolution of the source data. Neighborhoods that are smaller than the original data enumeration unit and lie completely within it will always produce the same mean center of population, as the distribution inside the enumeration unit must be assumed to be uniform. Neighborhoods that are too large will begin to tell a less localized story and each point on the output grid will have more similar values as the neighborhood size increases.

I decided on an initial neighborhood size of 55km for the first run of the Central Midwest study area. The approximate average area for all the enumeration units in this study area is 2,061 square kilometers. The square root of this gives us a value of about 45; I added 10 to this just to ensure the neighborhoods would be a little larger than the average enumeration unit. I also experimented with some other neighborhood sizes, but I will discuss the results of these in further detail in Chapter 4.

Calculating Change in Mean Centers of Population: Theory

At this point we have a matrix of point locations (in practice two matrices, one for the x coordinate, and one for the y coordinate) representing the focal mean population centers for a given year. In order to analyze population change through time, we must have the same data for

at least one other point in time. Therefore, this process must be repeated for each date layer using the *exact same neighborhood parameters*. Changing neighborhood sizes or shapes from one year to the next will obviously yield ineffectual results.

Once we have the focal mean centers of population matrix for two years, we can calculate a “difference” matrix. This is a matrix that records the changes between two years. The cell values in the “difference” matrix are actually vectors. The vectors’ magnitudes are the Euclidean distances between the population centers we calculated in the previous step. For instance, our first neighborhood example (the evenly distributed example) produced a population center at (3, 3) for its focus pixel. The second neighborhood example produced a center at (2.2449, 1.8776) as described above. For the sake of illustration, let us assume the first example is part of a year 1900 matrix, and the second is part of a 2000 matrix, and both neighborhoods cover the exact same geographic region.

In Figure 2.13, the green X is the population center of the neighborhood in 1900 and the red X is in 2000. If we connect the Xs—which we’ll name A and B for the sake of discussion—we end up with a vector **AB**. The magnitude of this vector is the Euclidean distance between these two points, and is calculated as such:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(2.2449 - 3)^2 + (1.8776 - 3)^2} = 1.3528$$

To calculate the direction, we make use of MATLAB’s *atan2* function, which calculates the four-quadrant inverse tangent. This function takes two parameters, an *x* value and a *y* value and calculates the direction in the $-\pi$ to π (-180° to 180°) range from the origin to the (*x*, *y*) point.

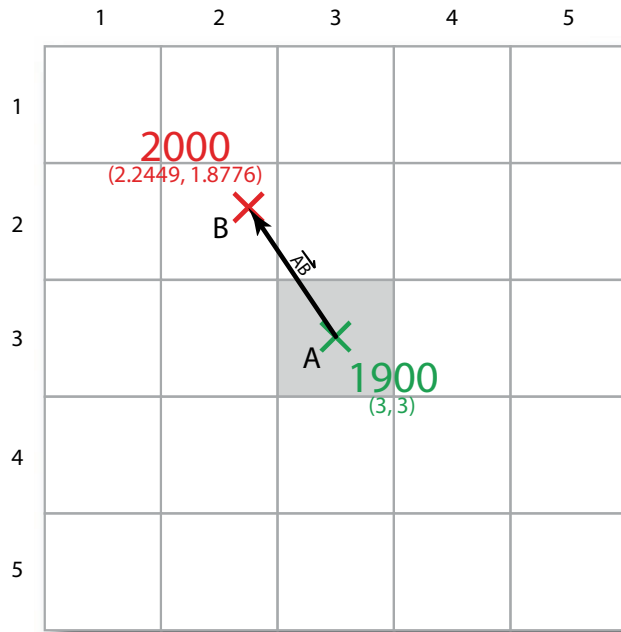


Figure 2.13. Mean centers of population for two years and the vector from one to the other.

For this to work properly, we must “reset” the first point (point A in Figure 2.13) to the origin, that is, we need to offset both points A and B so that point A locates at (0, 0) in Cartesian space and point B remains at the same distance and direction from A. It is extremely important to remember that the y axis in our matrix coordinate system runs in the opposite direction than that of the Cartesian coordinate system: y increases in a downward direction instead of upward as in the Cartesian grid. For x we can simply subtract the x value for the first point (point A) from both points. This zeroes out the x coordinate for point A. For y , we must subtract the y values *from* the y value of point A to zero out the y coordinate for point A. For our example above, for x we have $x_1 - x_1 = 3 - 3 = 0$, and $x_2 - x_1 = 2.2449 - 3 = -0.7551$. For y we have $y_1 - y_1 = 3 - 3 = 0$, and $y_1 - y_2 = 3 - 1.8776 = 1.1224$. We then take the $\text{atan2}(1.1224, -0.7551)$ — the MATLAB syntax for this function asks for the y value first—which produces the result: 2.163. This is an angle in radians (2.163 radians \approx 124 degrees) starting from due east and traveling

counterclockwise, ending up pointing in a northwestern direction as shown in Figure 2.14. The original vector **AB** has been added to the figure to verify the accuracy of the direction.

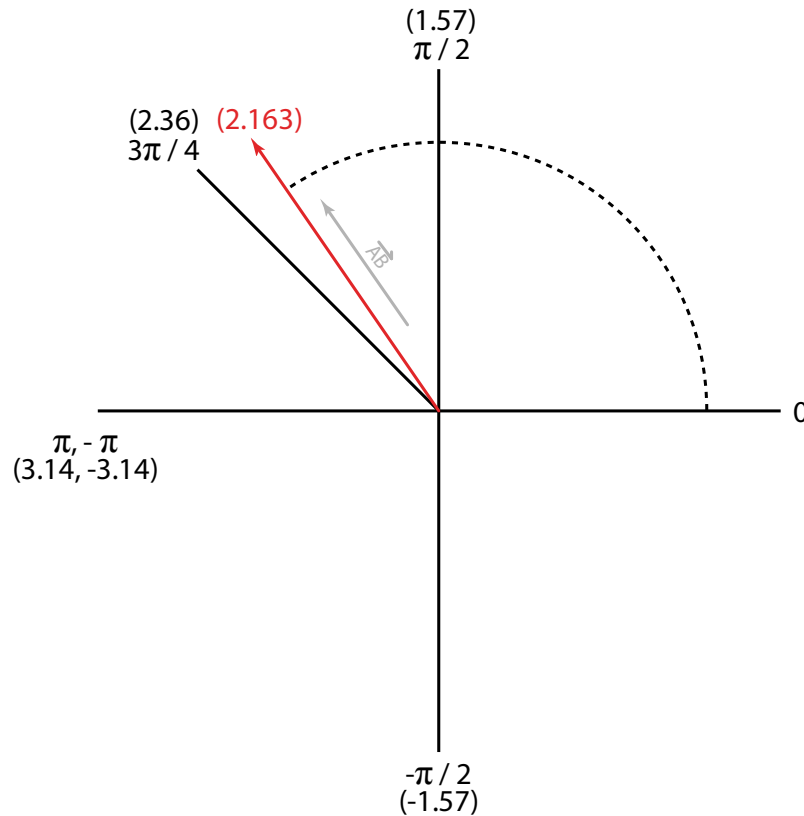


Figure 2.14. Results of the *atan2* function. The AB vector shown in Figure 2.13 is included for comparison.

With the *atan2* function, if the point is (0, 0), i.e, there was no change in the mean center of population, then there is no direction from the origin, and so this is the only case where the function will fail. If this occurs, MATLAB assigns a value of zero to the output cell.

Unfortunately, this zero is fundamentally different from the zero that would be returned for a due easterly direction, but as the magnitude would be zero at this point, this is irrelevant. If the point lies on the positive *x* axis, the direction is 0 (or due east); if it lies on the negative *x* axis, it is π (or due west). Similarly a point on the *y* axis will be due north or due south ($\pi/2$ or $-\pi/2$)

depending on its sign. The method of calculation of all other points depends on the quadrant it falls in, determined by the sign combination of x and y . By focusing only on the quadrant the point lies in, the angle from the origin to the point with respect to that quadrant's axes can be determined with simple right-triangle trigonometry.

In our example we have the x value -0.7551 , and the y value of 1.1224 , as described above. If we plot this in Cartesian space, we can create a right triangle as shown in Figure 2.15b. Since we know the lengths of the opposite and adjacent sides to angle α , we can determine the angle by taking the arctangent of $(0.7551 / 1.1224)$ which equals 0.5922 . Furthermore, since we want the total angle from the positive x axis, we simply add the 90° in the upper right quadrant ($\pi / 2$ radians, or 1.5708). So we have $\pi / 2$ or $1.5708 + 0.5922 = 2.163$, which is the final direction in radians as shown back in Figure 2.14. The algorithm in the computer function works a bit differently than how I described it here, but it ultimately achieves the same results.

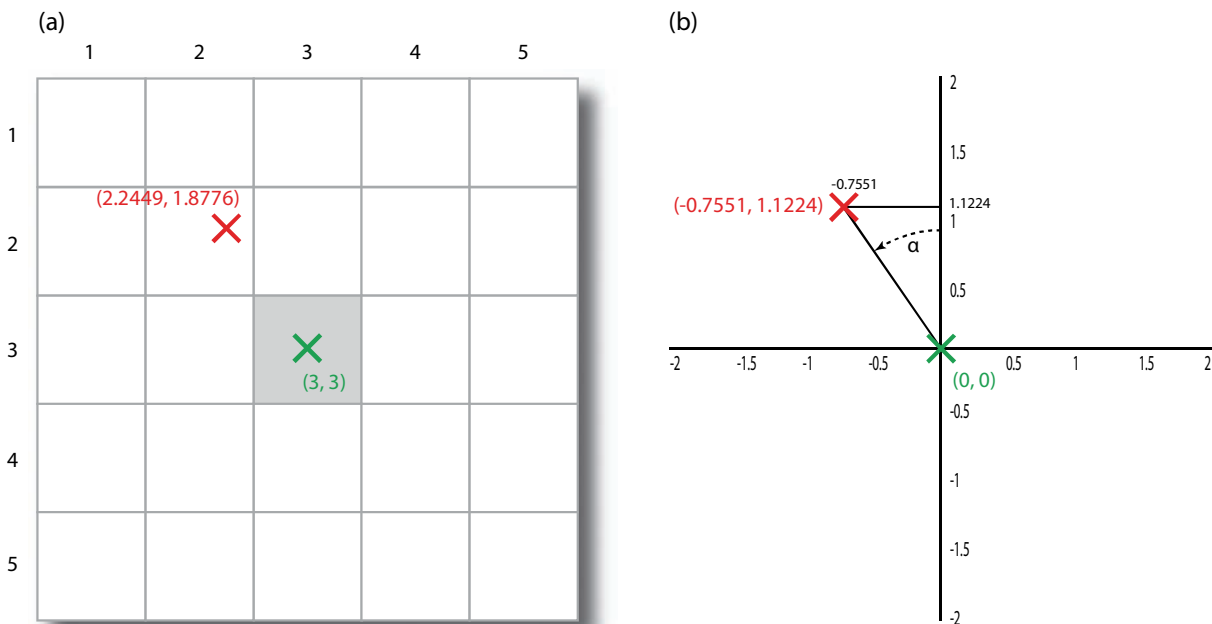


Figure 2.15. The data matrix coordinate system versus the Cartesian coordinate system and an example of the atan2 function.

Calculating Change in Mean Centers of Population: Implementation

I performed the calculations described above for each cell in the entire study area matrix, resulting in a vector for each cell in the matrix. In practice, this matrix actually consists of two matrices, one with the magnitude of the vector, and one with the direction of the vector. I wrote a pair of short MATLAB scripts to calculate the magnitude and direction matrices based on the four input matrices (1900 x values, 1900 y values, 2000 x values, and 2000 y values) calculated in the previous section. These are shown in Figure 2.16. They each accept as input the four matrices and generate as output the direction and magnitude matrices, respectively. These two matrices contain the components of our vector field, the symbolization of which I will discuss in Chapter 3.

```
function [dir] = direction(xgrid1, ygrid1, xgrid2, ygrid2)

% USAGE [dir] = direction(xgrid1, ygrid1, xgrid2, ygrid2) where the input
% grids consist of x and y coordinates of two different snapshots (1 and 2).
% The output is a single grid indicating the directional changes between
% snapshots in radians (-pi to pi range).

x = xgrid2 - xgrid1;
y = ygrid1 - ygrid2;

dir = atan2(y, x);

end

%-----

function [mag] = magdiff(xgrid1, ygrid1, xgrid2, ygrid2)

% USAGE [mag] = magdiff(xgrid1, ygrid1, xgrid2, ygrid2) where the input
% grids consist of x and y coordinates of two different snapshots (1 and 2).
% The output is a single grid indicating the magnitude changes between
% snapshots in positive floats.

mag = sqrt((xgrid1 - xgrid2).^2 + (ygrid1 - ygrid2).^2);

end
```

Figure 2.16. The *direction* and *magdiff* functions that calculate the direction and magnitude matrices.

Chapter 3

Visualizing Population Change Vector Fields

Creating a map that shows the vector field well is a difficult cartographic issue. In the simplest form, each vector in the field can be shown with its own vector symbol. This can be an arrow where the tail length represents the magnitude and the direction of the arrow represents the direction. If the vector field is small enough and the “cell size” large enough, this does not present an issue (Figure 3.1a). However, as the field grows larger and the “cell size” grows smaller, the arrows become too small to see (Figure 3.1b). One potential solution to the problem would be “resampling” the field, creating an arrow symbol which represents the mean magnitude and direction for a group of cells; however, this would be at the expense of the fidelity of the original data and result in a lower resolution field.

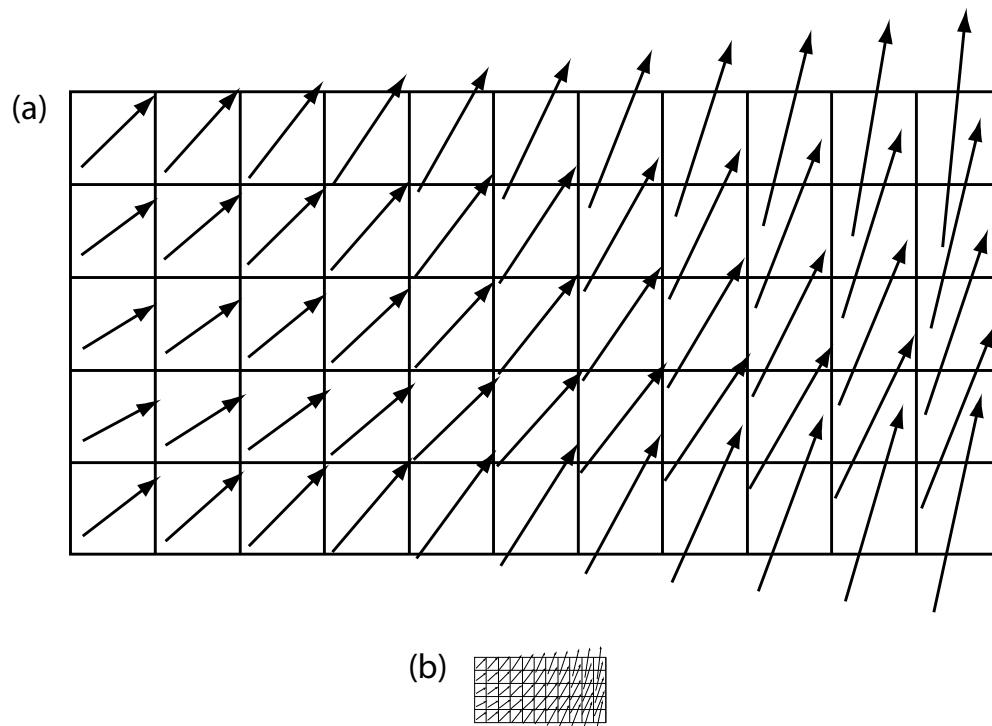


Figure 3.1. An example of using arrows to symbolize a vector field at two different scales. The directions and magnitudes of the arrows in (a) are clearly visible, while those in (b) are not.

To maintain maximum spatial resolution, we can abandon the use of arrows altogether and use color to represent the direction and magnitude of vectors. The HSV (hue, saturation, value) color model (Travis, 1991) offers a strong framework for symbolizing the two vector components. For example, Boring and Pang (1996) adopted the HSV color model to visualize the directional flow of vector fields using “direction-to-hue illumination”. The direction component of the vector is a circular variable—it has no minimum or maximum value in the strictest sense, as there really is no “starting point” or “ending point” when it comes to direction. Humans have elected to use the direction of north as a directional standard reference (such as north arrows on maps), but this is purely arbitrary. Two-dimensional directions are continuous around a compass rose and do not have a beginning or an ending. Similarly, the hue component of the HSV model is also circular—the lowest and highest values converge at the same shade of red (Figure 3.2).

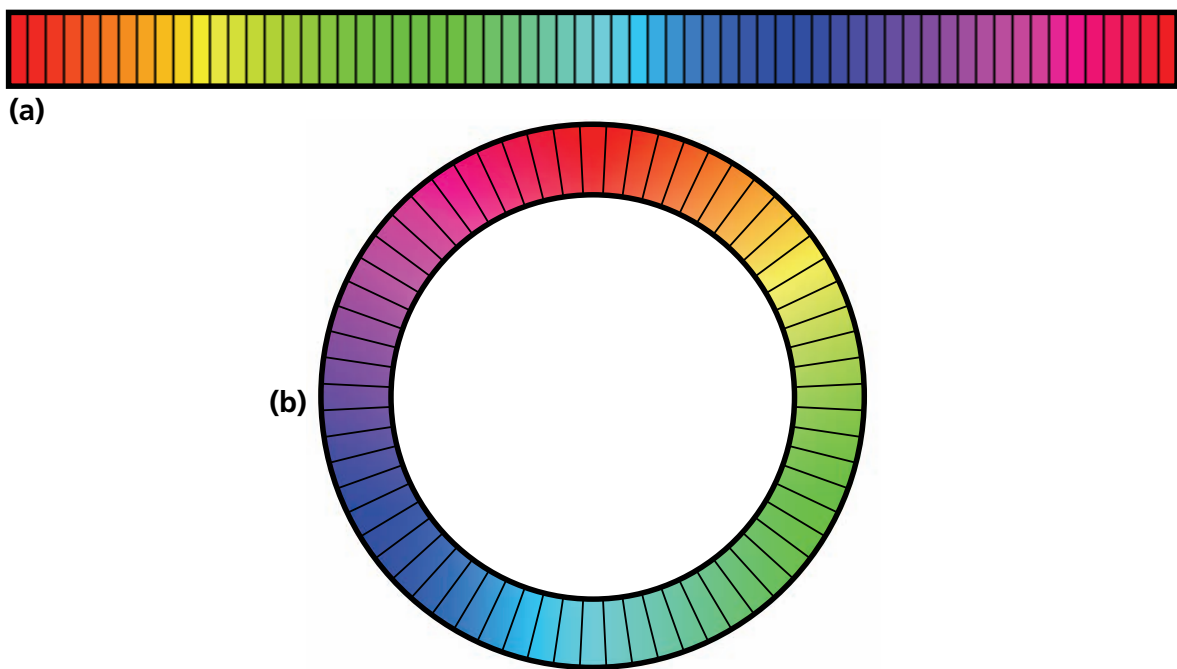


Figure 3.2. Hue band and hue wheel. The hue wheel (b) demonstrates how the hue bar (a) is continuous.

In contrast, the magnitude component of the vector is a positive scalar with potential values from zero to infinity. While the remaining saturation and value components of the HSV model are not limitless, the magnitude values that we are interested in portraying do not, in practice, approach infinity and can be scaled to fit in either the saturation or value component domains. In this chapter I will first discuss the symbolization of the direction component, followed by the magnitude component, and finally I will discuss the methods to combine the two using the HSV color model into a final composite map.

Association of Color with Direction

Since I am using the hue component of the HSV model to symbolize direction, I thought it would be a good idea to first determine whether or not there are any existing color-direction associations among potential viewers of the proposed symbology. In T.T. Waterman's article "Religious Practices of the Diegueño Indians," we learn that the tribe made associations of particular colors with the four principal directions based on songs and ceremonies in the tribe's culture (Waterman, 1910, p. 333). However, my research—as well as my general experience—has turned up no such associations in our modern culture.

Brewer and Marlow (1993) have done some work in using color to depict slope and aspect, building on the earlier work of Moellering and Kimerling (1990). They attempted to perfect an appropriate aspect coloring scheme—essentially assigning specific colors to each of the directions in aspect grids (raster datasets that show the direction of slopes). These choices were not based, however, on any existing cultural associations of color with direction.

Colors have taken on a number of meanings in various cultures, some of which are more universal than others. For example, most people will associate blue colors with cold temperatures and red colors with hot temperatures. In most cultures, green means "good" or

“go”, red means “bad” or “stop”, although this is not universal. The seasons of the year also often have colors associated with them. White and shades of blue are often associated with winter; greens are associated with spring; and browns, yellows, and reds with autumn. The cardinal directions, however, do not appear to enjoy any ingrained color associations. In the absence of any prior research into whether individuals have or do not have any pre-existing ingrained association of color with direction, I designed a simple survey.

The premise of the survey was to determine what associations the respondents make between eight distinctively different colors and a number of concepts—including direction. Choosing the eight colors was initially problematic because most color wheels seem to be divided into multiples of ten. Pulling out eight evenly spaced colors was not a straightforward task. Deriving colors from a standard hue wheel at 45° increments yielded eight “evenly spaced” colors; however, these colors were not distinctly different. There were still shades of color—particularly greens and blues—that were difficult to distinguish from one another.

The answer to this problem lies in opponent process theory. Using a syntactical relationship model originally developed by D. J. Cuff in 1972 and presented in J. Ronald Eastman’s article on opponent process theory (Eastman, 1986), I was able to derive eight distinct colors and create them in the RGB (red, green, blue) color model. I chose RGB because the survey would be conducted online, and an additive color model is optimal for this. The eight colors I chose along with their respective RGB values are shown in Table 3.1.

I developed the actual survey using Adobe Flash software. I kept the interface simple so as not to distract the respondent from the task at hand. The respondents were presented with eight color “nodules” (movable circles depicting each of the colors in Table 3.1), randomly arranged in two columns of four colors as shown in Figure 3.3. The original plan was to present

these nodules to the respondent and have him or her associate them with certain concepts, beginning with the most obvious associations and working toward the more abstract.

Table 3.1. Opponent process color choices and their respective RGB values.

Color	Red (R)	Green (G)	Blue (B)
■ Red	255	0	0
■ Orange	255	180	0
■ Yellow	255	255	0
■ Light Green	0	255	0
■ Dark Green	0	128	0
■ Blue	0	0	255
■ Cyan	0	255	255
■ Magenta	255	0	255

Table 3.2 outlines the concepts and the order in which they were originally going to be tested. I chose to include these concepts so that the respondent would not be aware of which of the six “quizzes” was of primary interest.

Table 3.2. Color association concepts (least abstract to most abstract).

- 1 STOP - GO
- 2 HOT - COLD
- 3 WINTER - SPRING - SUMMER - FALL
- 4 MORNING - NOON - EVENING - NIGHT
- 5 LEFT - RIGHT
- 6 N - NE - E - SE - S - SW - W - NW

I also hoped that it would be easier to identify genuine responses as opposed to random responses by examining the responses of the less abstract concepts. For example, we could potentially conclude that the respondent is not taking the exam seriously if red and green nodules were not associated with “stop” and “go” respectively. Unfortunately, the fact that I have little experience with ActionScript—the proprietary scripting language used in the Adobe Flash

environment—forced me to abandon the concept progression, and instead I only administered the portion covering the sixth concept of direction.

Each respondent was presented with an empty compass rose and eight randomly placed color “nodules” and was directed to place the nodule that he or she felt best represents the direction into the empty “node” in the compass rose (Figure 3.3). I designed the survey to take no more than one or two minutes. To tackle the inevitable bias that may be produced by colorblind individuals, I provided a checkbox for those who knew they had a colorblind condition, so these responses could be removed from the results if desired. Finally, I added a “reset” button that would return all the color nodules to their original positions in the event that the respondent wished to start the survey over from scratch. To ensure that each color nodule was placed into one of the eight color nodes, the “finish” button did not function until each node had one—and only one—nodule in place.

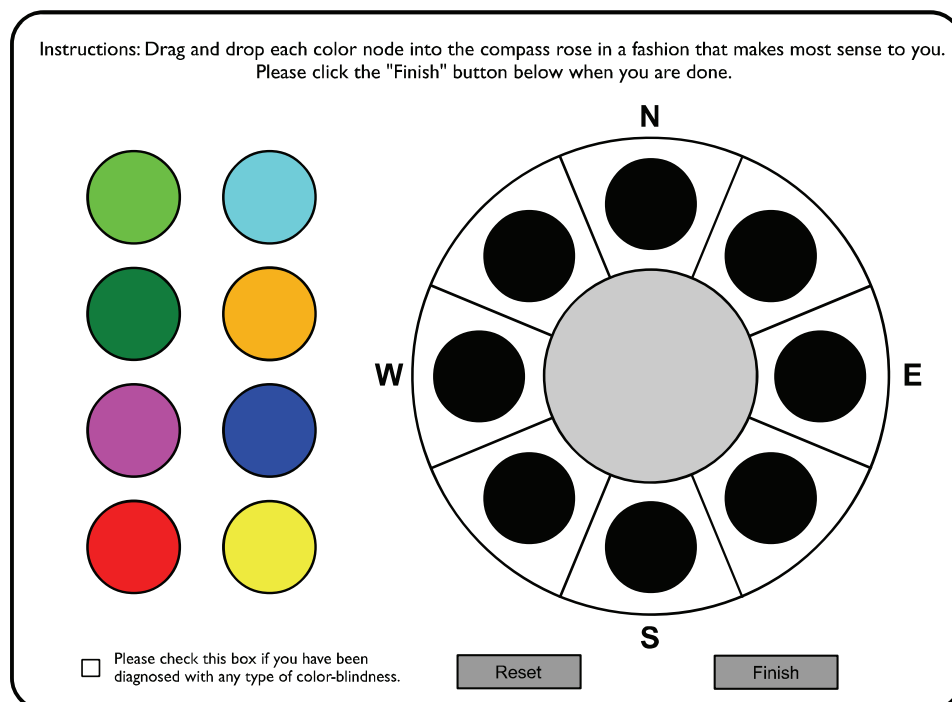


Figure 3.3. The online color survey interface.

To make the quiz easier, I programmed the nodules to “slide” into the node receptacles and make a reassuring clicking sound as long as the nodule was reasonably close to the receptacle and provided the receptacle was not already occupied by another nodule. This required quite a bit of coding to perfect. The ActionScript 3.0 code for the survey is shown in Appendix C. Another challenge was making it possible to save the results to a file for later retrieval. To accomplish this I had the ActionScript code send a URL (Uniform Resource Locator) request containing the results to a php script (shown in Appendix D), which broke down the URL structure and wrote the results to a text file on a web server.

My original hypothesis was that there would be no pattern in the responses to the eight direction elements. However, informal preliminary discussion with some of my colleagues indicated that most would select red or blue for north. When I asked why this was, most replied that red is the strongest color and north seemed to be the place to put it. Those who said they would select blue indicated that blue is associated with cold, as is the north (at least in the Northern Hemisphere).

I conducted the actual survey over a three-month period. In an effort to maximize the number of respondents, I asked several of my colleagues if they would offer extra credit of some sort to their students in return for taking the survey. Because of this, the vast majority of the respondents were most likely college freshmen or sophomores in the 18-20 year age range. To ensure that the students complied, I included a confirmation number that would be provided to the respondent at the end of the test. I then provided the results file with the list of numbers to the instructor who could compare the submitted numbers against those stored in the file. The raw results text file is shown in Appendix E.

Each of the survey responses is coded in the following format:

compass=0,6,7,5,4,3,2,1,8,0,1647

The “compass=0,” portion is related to the method used to export the results through a URL, and is essentially a header to denote the start of a new response. The next eight digits indicate the colors and the locations they were placed in the compass rose. The colors are numbered in the same order that they are presented in Table 1 (i.e., 1=red, 2=orange, etc.). The directions are indicated by the positions in the series. A number in the first position means that number’s color was assigned to north. These positions continue clockwise until the eighth position which represents northwest. The last single digit is the colorblind bit (0 for no, 1 for yes), and the final number (4 or 5 digits) is the confirmation code.

When the survey was complete there were 141 respondents, three of whom indicated that they were colorblind. Since this represented a mere two percent of the respondents, I decided to eliminate those responses, leaving 138 valid responses to the survey.

Because the direction colors are dependent variables (the direction a color is assigned is dependent on whether or not a color has already been assigned that direction) there is no way to test for statistical significance. However, there is a clear pattern in these results that I did not expect in my hypothesis. Figure 3.4 shows the frequency of colors chosen for each of the eight compass rose directions. Figure 3.5 shows the results by color, showing the frequency of directions chosen for each. At the bottom of Figure 3.5 are the four configurations that were chosen most frequently by the respondents. The exact configuration shown with the number 22 was chosen by 22 of the respondents, which represents just under 16% of the people surveyed.

The configuration chosen by nine of the respondents is very similar to the one chosen by 22, differing only in the transposition of the two blue hues.

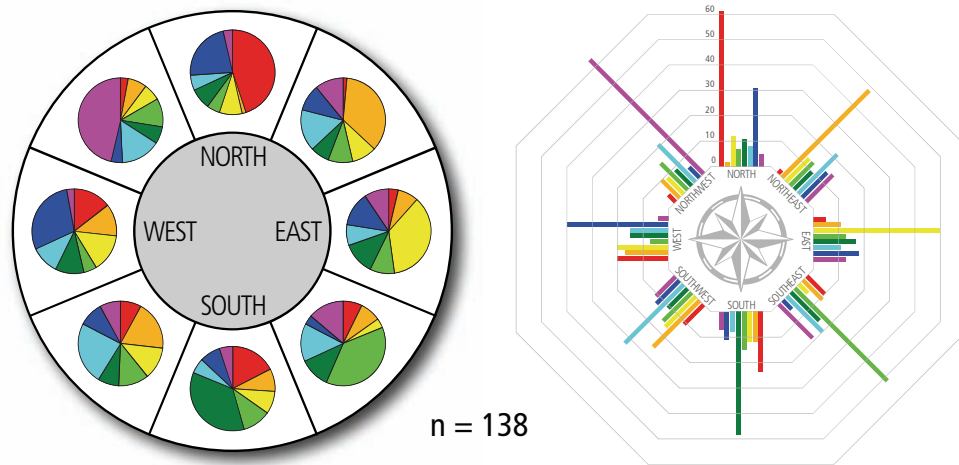


Figure 3.4. The color survey results by direction.

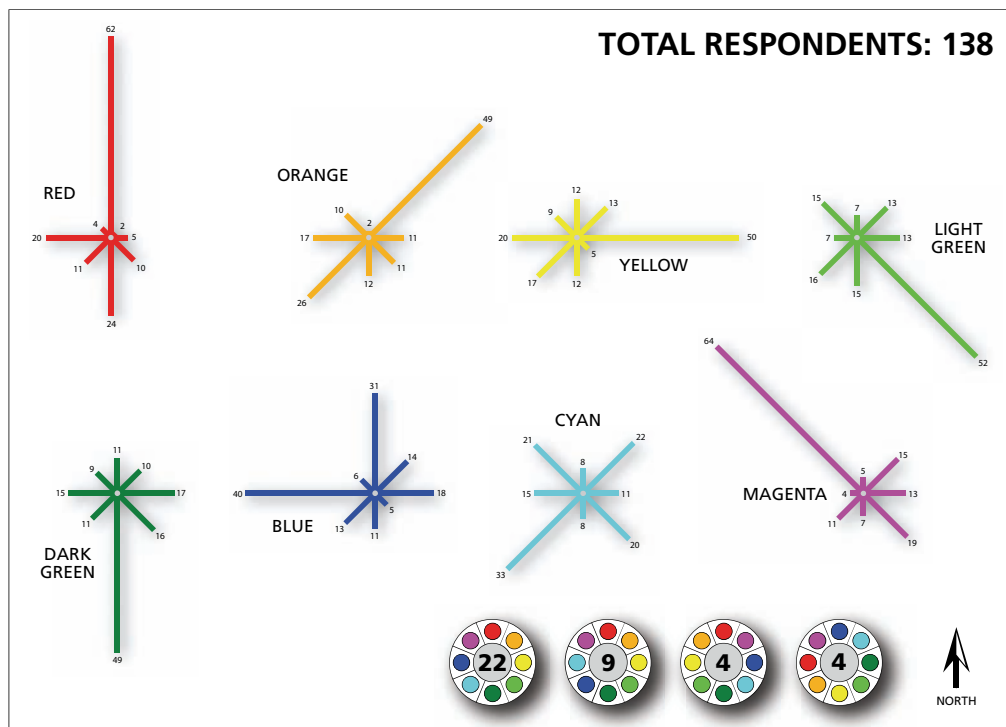


Figure 3.5. The color survey results by hue. Each of the eight star-shaped figures represents one of the eight colors. The frequency that the color was chosen for a given direction is indicated by the length of the bar pointing in that particular direction.

Furthermore, the two configurations chosen by four people are merely variations of the configuration chosen by 22 respondents—the first being the same configuration in a counterclockwise fashion and the other being the same configuration shifted 45 degrees. There were two configurations chosen by three people and nine chosen by two people. The remaining 75 configurations were unique.

While the results showed some patterns (nearly 45% of the respondents chose red for north while 22% chose dark blue), I suspect that in many cases, once a color was selected for a particular direction, there was likely a tendency to “create a rainbow”, automatically filling in the adjacent directions with the adjacent colors in the rainbow. While this is not necessarily a problem, it is certainly an issue, as only one direction is really being directly considered for match-up with a color, while the rest are assigned colors on the basis of the first color chosen. That aside, the “rainbow” assignment of colors is actually quite practical, as it somewhat matches the hue wheel shown back in Figure 3.2b.

In theory, the visualization of the vector fields would utilize the color system chosen by the plurality of respondents. If we look at Figure 3.6, we can see that the hue wheel is similar to the most popular pattern with a few important differences.

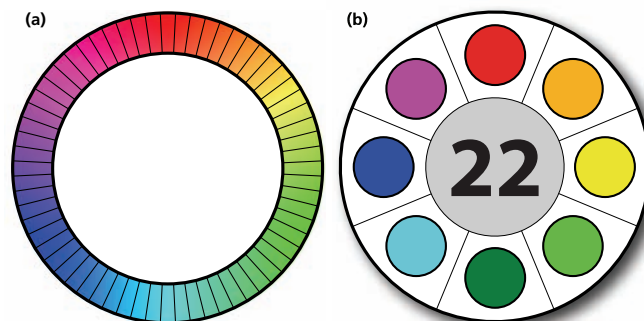


Figure 3.6. The HSV hue wheel and the plurality result of the survey (the exact pattern chosen by the most respondents).

We can see that north is indeed red and northeast is orange, however as we approach the easterly direction the hue wheel gets into the greens while the east node of the plurality result is clearly a yellow. Furthermore, the dark green color in the south node is not on the hue wheel at all. This particular color would require adjustment to the value component of the HSV color model to generate. These issues aside, and with the numerous responses coming reasonably close to the HSV hue wheel, I made the decision to use that hue scheme for symbolizing the direction component of the mean change vectors.

Symbolizing the Direction Component of a Vector Field

Application of the HSV hue wheel to the directional component matrix that we calculated in Chapter 2 produced the result shown in Figure 3.7.

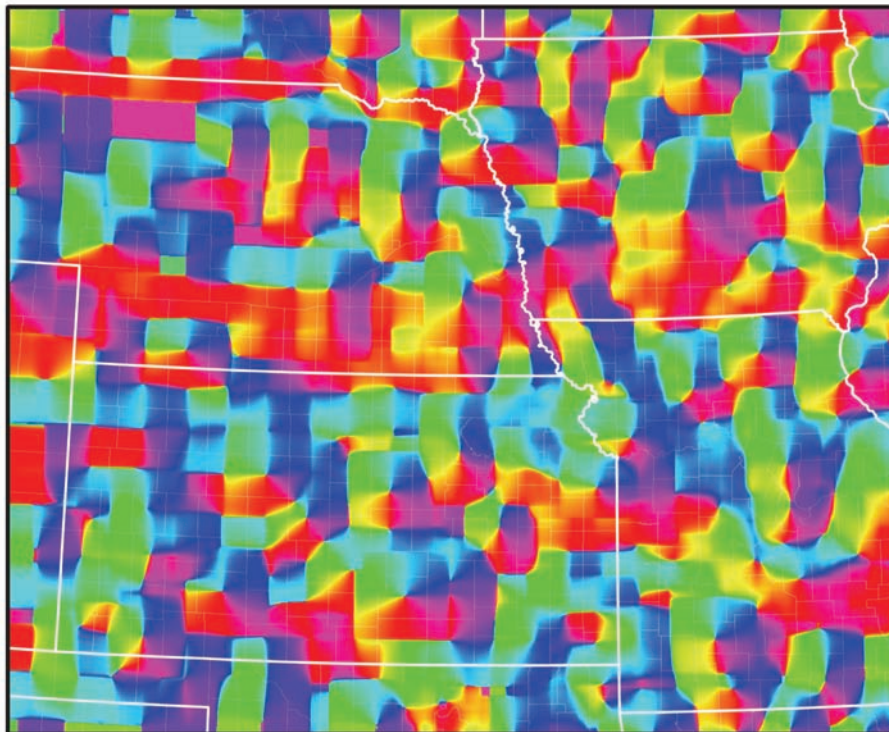


Figure 3.7. Direction of change in mean population center from 1900 – 2000 in the Central Midwest study area. The directions are mapped using the HSV hues.

The original matrix had a range of values from $-\pi$ to π (-180° to 180°) and needed to be stretched to match the hue component value domain ($0 - 1$), as well as “flipped” to match the clockwise geographical orientation as opposed to the counterclockwise orientation of the Cartesian system. To accomplish this, I used *dir2hue* function (Figure 3.8).

By adding π to all the values in the *dirgrid* matrix (the output grid from Chapter 2) we create a new matrix with a possible range of 0 to 2π . Dividing those values by 2π yields a grid with possible values of 0 to 1 , the proper domain for the HSV component grids. Because the Cartesian coordinate system starts in the east-west axis and the HSV hue wheel starts with the red hue, I needed to shift the hue values by 90 degrees ($\pi / 2$) to ensure the red hue was in the north position, as discussed earlier. I also needed to “reverse” the direction of the angle measurements; the Cartesian angles increase in the counterclockwise direction, while geographic angles increase in the clockwise direction. I accomplished this by subtracting the result above from 0.75 which “inverts” the values (changes increasing to decreasing) and “rotates” them by 90° . This produces some negative values, which, when added to 1 “fall back” into place in the $0.75 - 1$ range. Finally, I mapped the output grid to the HSV color ramp (as shown in Figure 3.2a) and overlaid the state and county boundaries for reference (I will discuss the procedure for doing this in Chapter 4).

A number of interesting patterns emerge in Figure 3.7, although it is important to note that without the magnitude component figured into the visualization, the importance of the patterns is as yet indeterminate. Areas of zero magnitude are still represented here with a hue, even though the direction in those cases is technically undefined. This occurs because the MATLAB calculations that should have resulted in a value of 0 instead resulted in extremely small numbers (in the $1.0\text{e-}13$ range). I have been unable to determine the cause of this, but it is

largely irrelevant—for the purposes of symbolization, these values are, for all intents and purposes, treated as zeroes. Once we incorporate the magnitude into the composite, the hues in these locations will not be visible, as areas with zero magnitudes will simply be symbolized in black.

```
function [huegrid] = dir2hue(dirgrid)

% USAGE [huegrid] = dir2hue(dirgrid) where dirgrid is a double float grid
% with a range of values from -pi to pi. The output is a double float grid
% with values from 0 to 1.

hg = 0.75 - ((dirgrid + pi) ./ (2*pi));
huegrid = hg;
huegrid(hg<0) = hg(hg<0) + 1;

end
```

Figure 3.8. Function to calculate the hue grid from the direction grid.

Symbolizing the Magnitude Component of a Vector Field

The magnitude component symbolization is somewhat more straightforward than the direction component. The value component of the HSV model is a measure of the brightness, or the whiteness/blackness of the hue (Travis, 1991 pg. 80). The low values will create darker hues, the higher values will create lighter hues. Values of zero will completely drown the hue in blackness and the result will be pure black. Values of one will display the hue just as it is shown in the HSV hue wheel. By applying these dark / light values to the magnitude component, the closer the magnitude gets to zero (where the direction is undefined), the less hue you will see.

The primary hurdle is stretching the magnitude values to fit into the value component range (0 – 1) in the HSV model. At the simplest level, the magnitude values can be converted to the value range in a linear fashion. As all the magnitude values are positive and the minimum magnitude value is assumed to be zero, simply dividing each magnitude measurement by the maximum measurement in the matrix will produce a linear stretch relationship in converting

magnitude to the value component range (0 – 1). The blue line in Figure 3.9 demonstrates a linear stretch. The problem with this method is that many of the vectors with lower magnitudes will be “drowned out” by black and will be barely visible, if visible at all in the composite. Notice how dark the linear magnitude bar at the bottom of Figure 3.9 is at the left end where the lower magnitudes reside.

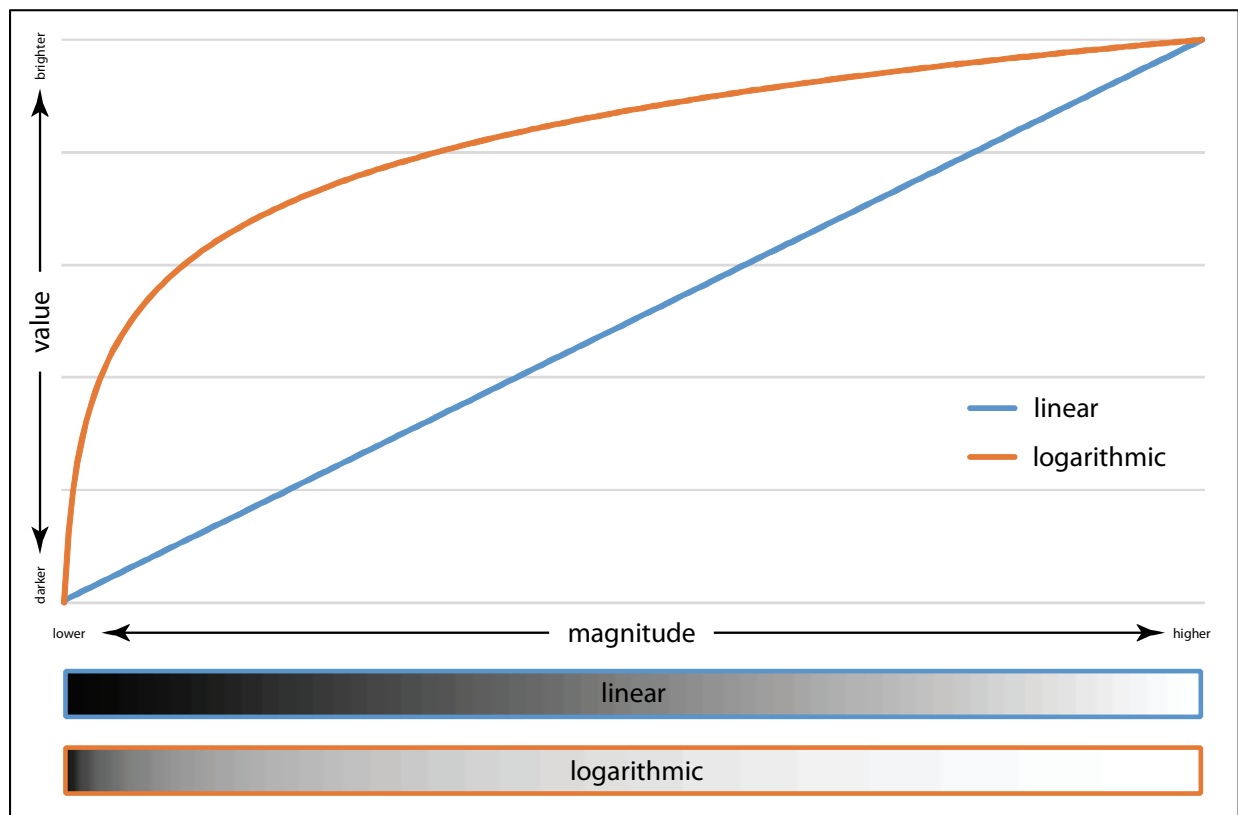


Figure 3.9. Linear vs logarithmic stretch of the magnitude of vectors to the value component range (0–1) in HSV.

We can alleviate this issue to a degree by employing a logarithmic stretch to the magnitude values, as shown with the orange line in Figure 3.9. As you can see, the lower magnitudes are not quite as dark, and the hues for the lower magnitude vectors will still be visible in the composite. Whether we choose the linear or logarithmic magnitude scaling is largely dependent on what we wish to focus on in the composite map we generate. If the interest

is only in higher magnitude vectors, the linear would be more appropriate. If you wish more of the lower magnitude vectors to be visible, the logarithmic would be the better choice.

I wrote the *mag2val* MATLAB function shown in Figure 3.10 to convert the magnitude grid (*maggrid*) that was calculated in Chapter 2 into the value component grid (*valgrid*) for the HSV composite. This function accepts two inputs, the *maggrid* magnitude grid and a flag value for which of the two scaling techniques we wish to employ. Passing a 0 to the function will generate a grid with linear scaling, while passing a 1 will generate a grid with logarithmic scaling.

```
function [valgrid] = mag2val(maggrid, linlog)

% USAGE [valgrid] = mag2val(maggrid, linlog) where maggrid is a double float
% grid with positive double float values and linlog is 0 for a linear scaling
% of magnitude or 1 for a logarithmic scaling of magnitude. The output (valgrid)
% is double float grid with values from 0 to 1.

if (linlog)
    grid = log(maggrid + 1);
else
    grid = maggrid;
end

maxpos = max(grid(:));
minpos = min(grid(:));
rngpos = maxpos - minpos;

valgrid = ((grid - minpos) ./ rngpos);

end
```

Figure 3.10. MATLAB function to convert magnitude grid to value grid.

The linear scaling works by simply subtracting the minimum value in the entire grid from each grid value and dividing that result by the range of the grid (the maximum minus the minimum), generating the result shown in Figure 3.11a. The logarithmic scaling uses the *log* function to generate a logarithmic curve that has a variable slope, starting very high and

becoming lower as the values increase. The result for the logarithmic scaling is shown in Figure 3.11b.

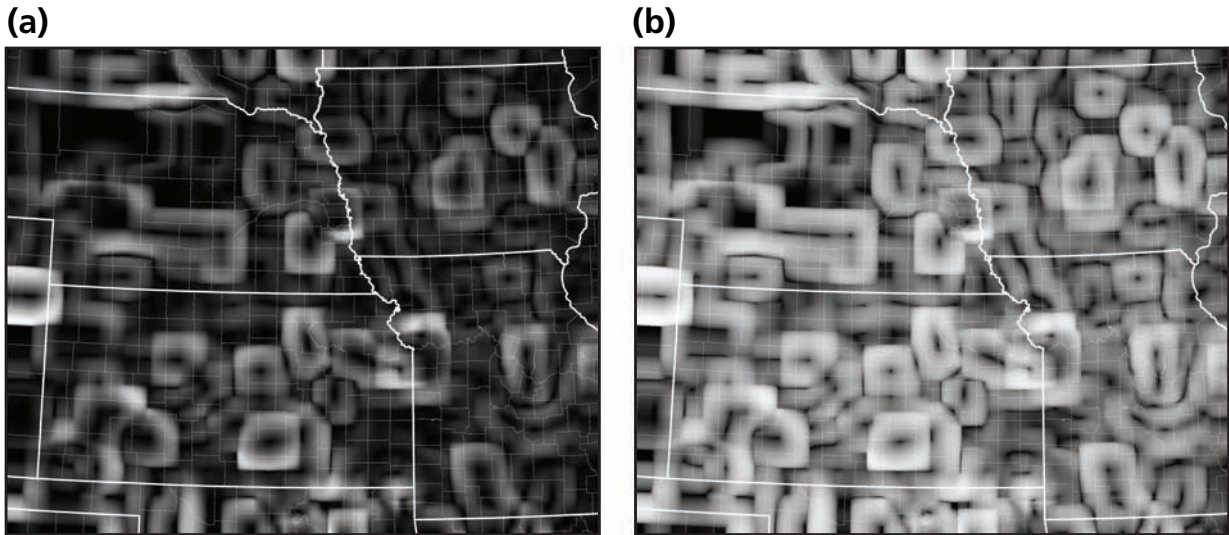


Figure 3.11. Magnitude of change in mean population center of the Central Midwest study area, 1900 – 2000, linear scaling (a), and logarithmic scaling (b).

Generating the Direction-Magnitude Composite Map

In the previous sections we have determined that the direction component will be mapped to the hue component of the HSV color model and the magnitude component will be mapped to the value component. The remaining component, saturation (S) could, conceivably be used to represent a third variable; however, as we are only concerned with the two components of the vectors, I have chosen to simply set the saturation component to a constant value of 1 across the entire grid.

Prior to working with the actual vector field data from Chapter 2, I generated three test matrices to represent each of the HSV color components. The hue matrix increases in value in a linear fashion from 0 – 1 from left to right, while each column remains constant. The saturation

matrix is set to a constant value of 1 over the entire matrix, and the value matrix increases in value in a linear fashion from 0 – 1 from top to bottom with each row remaining constant. With these three components in hand, I executed a routine to convert these to the respective RGB components so they could be combined into an RGB composite image. The MATLAB function *hsv2rgb*, largely based on the conversion algorithm outlined in Travis’ (1991 pgs. 212-213) text converts the HSV matrices to the RGB matrices. The input matrices and resultant RGB matrices are stylized in Figure 3.12.

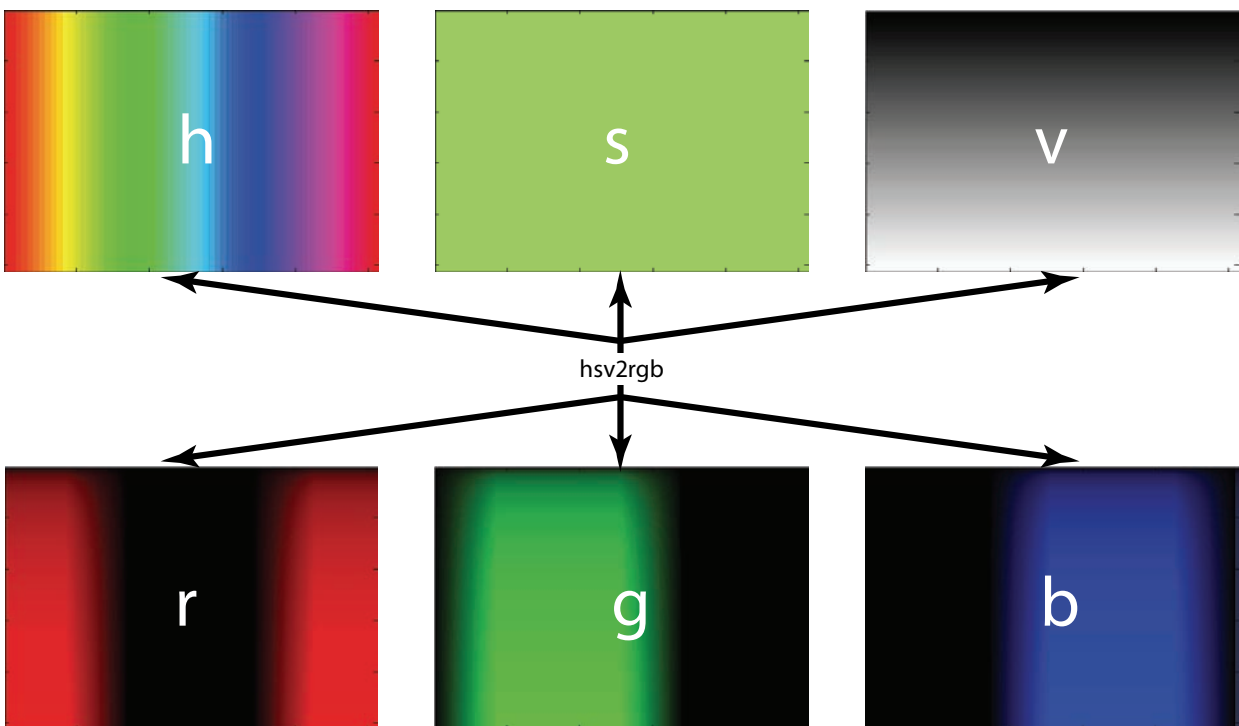


Figure 3.12. The three test HSV component layers and their RGB conversions.

Figure 3.13 shows final composites of the test HSV components. The directional values (the hue component) vary from left to right, the magnitude values (the value component) increase from top to bottom and the unused saturation component is constant across the grid.

Notice the difference in the obscured hue areas near the top of the figures between the linear (left) and the logarithmic (right) scaling of the magnitude component.

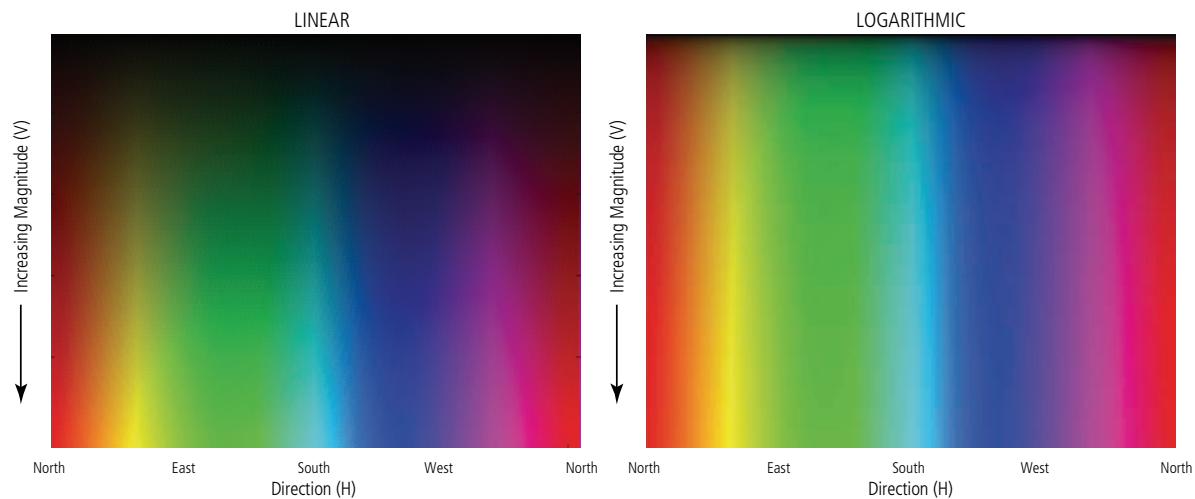


Figure 3.13. Linear (left) and logarithmic (right) test composites.

With the successful outcome of the test grids, I combined the direction/hue and magnitude/value grids of the study area with a constant saturation component of 1 and ran these through the HSV to RGB conversion. Next, the R, G, and B component matrices were combined using the MATLAB *cat* function into a three-dimensional array, where the third dimension consists of the R, G, and B components. Finally, the three-dimensional array was exported from the MATLAB format to image (TIFF) files using the MATLAB *imwrite* function. The linear magnitude result is shown in Figure 3.14 with the state and county border overlays added for reference. The logarithmic result is shown in Figure 3.15.

Streamlining the Process

Repetition of the individual steps described here and in Chapter 2 for multiple datasets is easily automated. All of the MATLAB functions I have described can be combined into a single function that handles the intermediate steps with complete transparency to the user. The *cimc*

function accepts the two population density raster grids (calculated in the pre-processing section of Chapter 2) as input and produces the final three-dimensional RGB composite. All that is required are the pre-processing steps (as shown in the flowchart in Appendix A) and then the execution of the *cimc* function on the two pre-processed raster grids. The flowcharts shown in Appendix B and F are combined into the *cimc* function as shown in Appendix G. The complete code for this function is shown in Appendix H.

In Chapter 4, I will discuss some post-processing procedures to generate the final presentable mapping products. I will also discuss some of the issues and problems with the finished maps, and introduce another study area with a few different parameters to explore these further. I will also assess the final products' legibility and determine how they are being interpreted by administering an assessment survey and discussing the results.

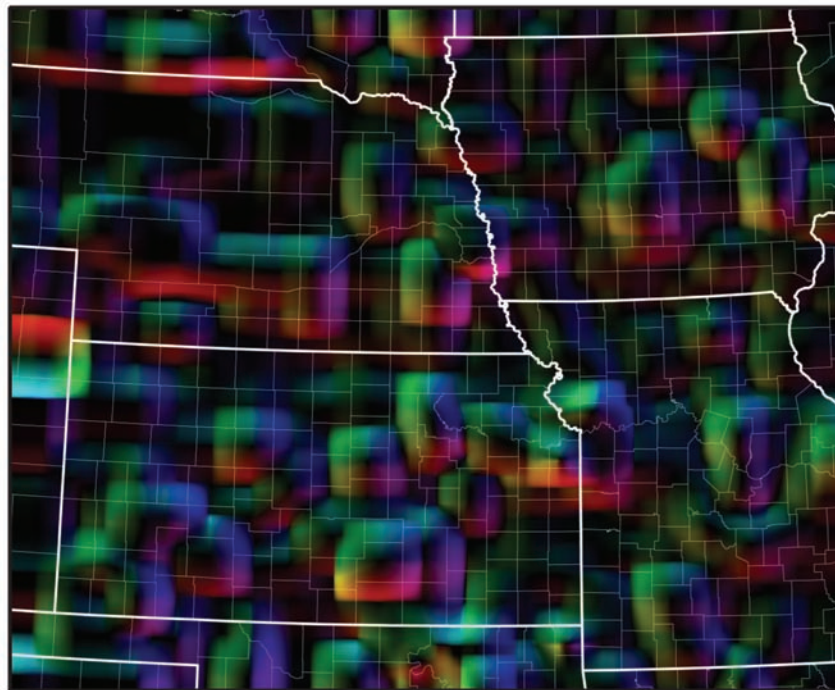


Figure 3.14. Composite map of population change between 1900 and 2000 with a neighborhood size of 55km, and using a linear magnitude stretch.

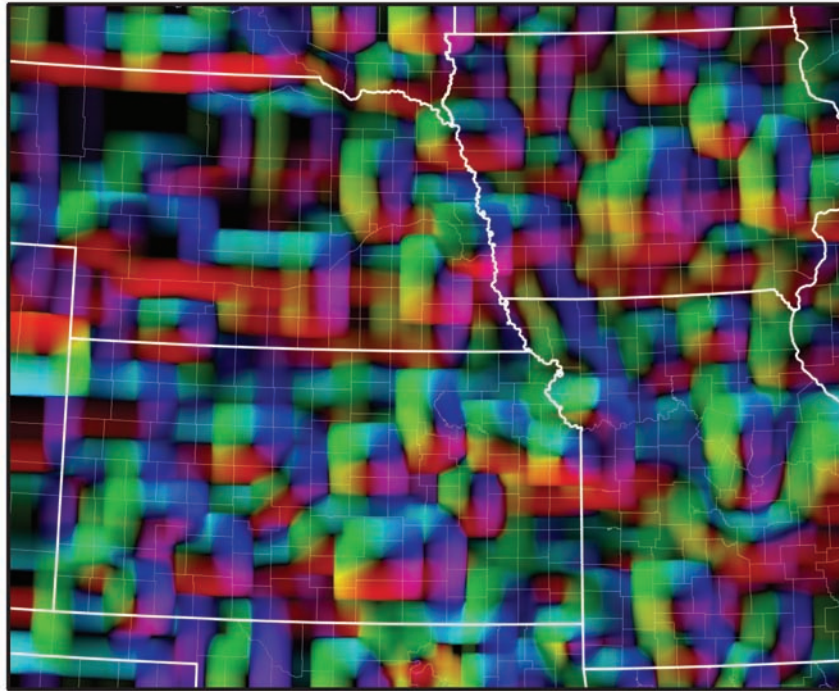


Figure 3.15. Composite map of population change between 1900 and 2000 with a neighborhood size of 55km, and using a logarithmic magnitude stretch.

Chapter 4

Presentation and Assessment

In Chapter 2, I generated a vector field over the Central Midwest study area based on the changes in mean center of population in 55km moving windows from 1900 to 2000. Because of the visualization problem with symbolizing the vector field in a traditional fashion (as shown in Figure 4.1), I determined that an alternate symbolization technique would be required.

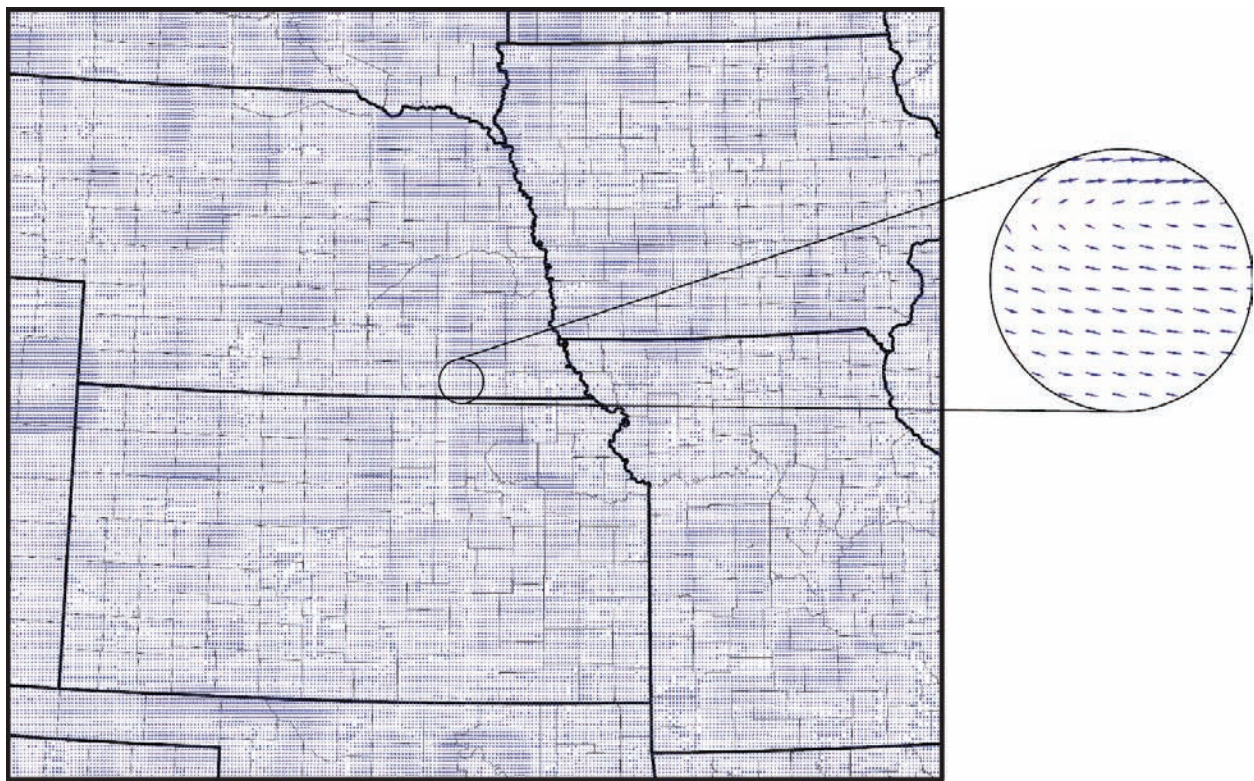


Figure 4.1. Central Midwest study area with traditional (arrow) symbolization of the change in mean center vector field (1900 - 2000).

In Chapter 3, I developed a pair of composite maps (one with a linear magnitude scaling, the other with a logarithmic scaling) of the study area using the HSV color model to symbolize the vector field. Figure 4.2 illustrates the relationship between a vector arrow symbology and its corresponding HSV color symbology. The three sample vectors on the left are placed in the

color wheel on the right with their origins in the center. The hue and value combination at the end of the arrow is the color symbol for that particular vector. In this chapter, I will discuss the procedures required to make these maps more useful, including the application of overlays for context and a legend that describes what is depicted on the map itself. I will also discuss the results, including some of the issues encountered with the Central Midwest study area and another study area, the Kansas City metropolitan area. Finally, I will discuss the administration of an assessment survey and its results.

Map Presentation

Figures 3.14 and 3.15 in Chapter 3 are the final results of the examination of the Central Midwest study area, but they are not polished. On their own, they will make no sense to the map reader; he or she will simply see a rectangle with many blotches of color. To remedy this, we must provide some context and some instruction as to what the map is symbolizing.

I have already included the state and county boundary overlays in the final maps from Chapter 3, but I did this only for the benefit of the reader of this thesis. The actual outputs from the *cimc* function do not include these. To include the overlays, I imported the TIFF file of the final RGB composite into ESRI ArcMap along with the NHGIS shapefiles that I simplified and clipped to the Central Midwest study area in Chapter 2. I then georeferenced the TIFF image using the study area corners as control points. Since there was no change in the extent of the study area or the coordinate system of the spatial data at any point in the data processing, the georeferencing was rather straightforward. Finally, I added a state boundary layer (downloaded from the NHGIS separately) for further context, and symbolized the boundaries in white to ensure there would be no clashing with the data on the map. Since the saturation component of

the HSV color model was kept constant at 1, it is impossible for any purely white areas to be part of the map symbolization. This leaves white as the optimal color for any overlays.

The overlay provides spatial context, but it still does not solve the problem of informing the map reader what the colors on the map mean. To do this, I needed to develop a legend that would be both intuitive and concise. I settled on generating a compass wheel legend that would indicate the hue around the perimeter for the appropriate direction and the value radiating from the center to the edge indicating magnitude, much like the graphic in Figure 4.2.

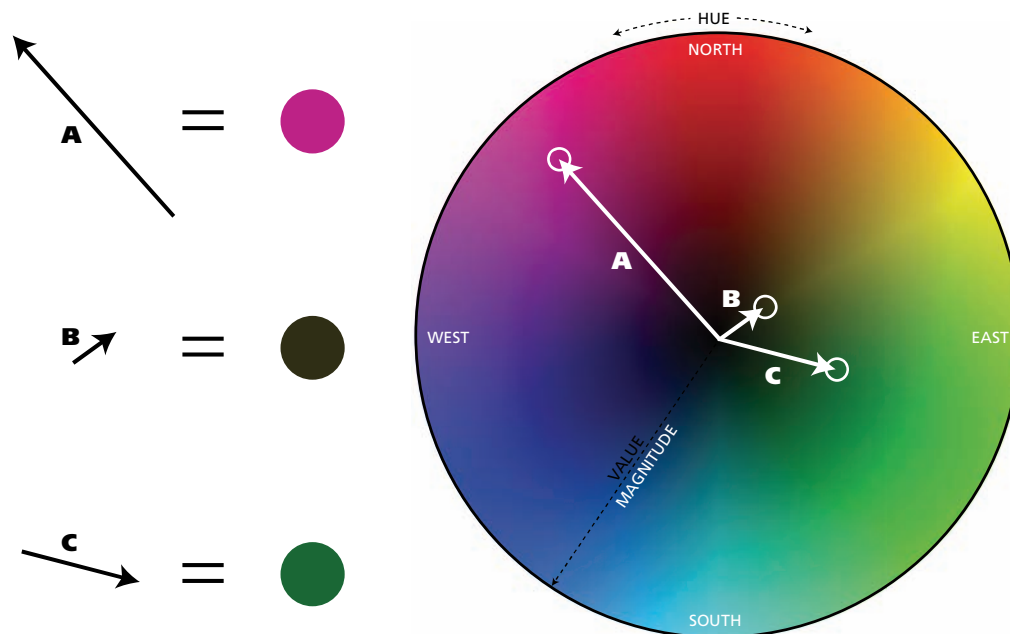


Figure 4.2. Translation of vector arrow symbology to HSV symbology. By placing the three example vectors (A, B, and C) onto the HSV hue-value construct, we arrive at the corresponding symbolization color.

Generating the compass wheel required the MATLAB code shown in Figure 4.3. I borrowed some portions of this code from a script posted to the MATLAB help forums (Image Analyst, 2015). This generates a 300 x 300 grid where the values increase from the center outward and the hue changes around the perimeter. The saturation is constant over the entire

grid. I imported the output grid into Adobe Illustrator to crop it to a circular shape and to overlay the directional labels. Combining the finished compass wheel with a gradient bar explaining the value-magnitude relationship and the original map resulted in the finished product (for the logarithmic scaling) as shown in Figure 4.4.

```
function [rose] = compass_rose_val()

% USAGE: [rose] = compass_rose_val()
% Generates a hue wheel with red at north going ROYGBIV clockwise
% Value increases from center of wheel to edge
% Saturation is constant

sz = 300; m = sz/2;

h = zeros(sz,sz);
s = ones(sz,sz);
v = zeros(sz,sz);

for i = 1 : sz
    for j = 1 : sz
        v(i, j) = min(1, sqrt((j - m)^2 + (i - m)^2) / m);
        h(i, j) = atan2((j - m), (i - m));
    end
end

h = fliplr((h + pi)/(2 * pi));
[r, g, b] = hsv2rgb(h, s, v);

rose = cat(3, r, g, b);

end
```

Figure 4.3. Code to generate the hue-value compass rose legend.

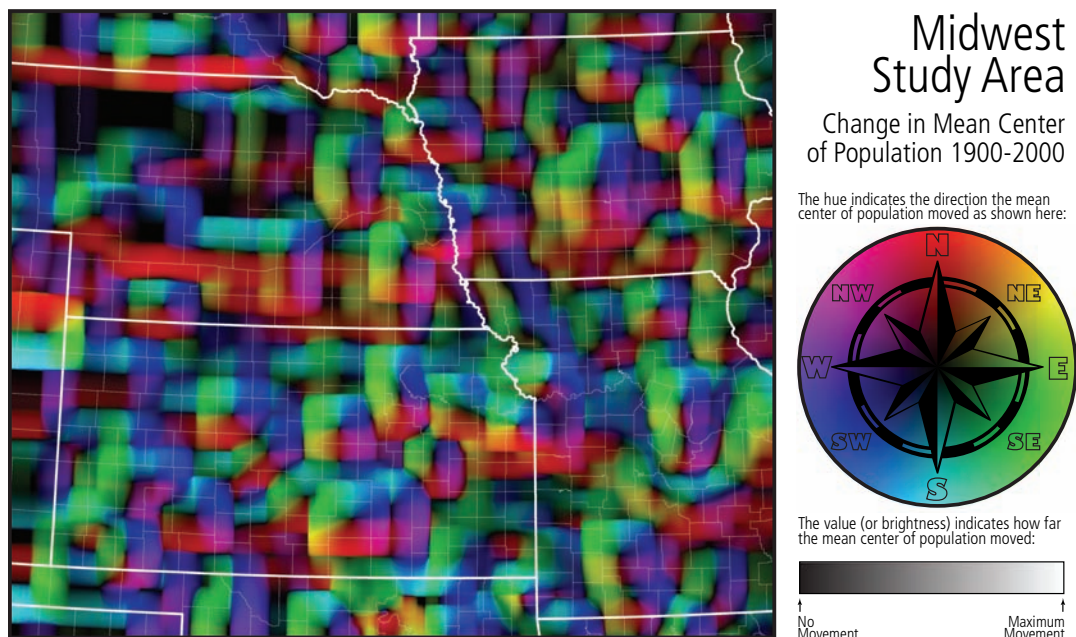


Figure 4.4. Final Central Midwest linear scaling product with overlay and legend.

Discussion of Results

Upon completing the first pair of maps using the 55km moving window over the Central Midwest study area, I began the process of trying to interpret the patterns in the map. The most striking patterns that stood out to me on first glance are the “donuts” of color found in certain parts of the map.

The two white circles in Figure 4.5 outline one type of these “donuts”. In these areas, the colors around the center are the opposite of the colors around the compass rose legend (reds are in the south, greens are in the west). This indicates that the mean centers of population gravitated toward the center of the “donut” shapes. These two convergence examples in the map correspond to the Des Moines metropolitan area (upper right circle) and the Wichita, Kansas metropolitan area (bottom center circle). Both of these areas saw considerable growth in population from 1900 to 2000 (United States Bureau of the Census, 2007).

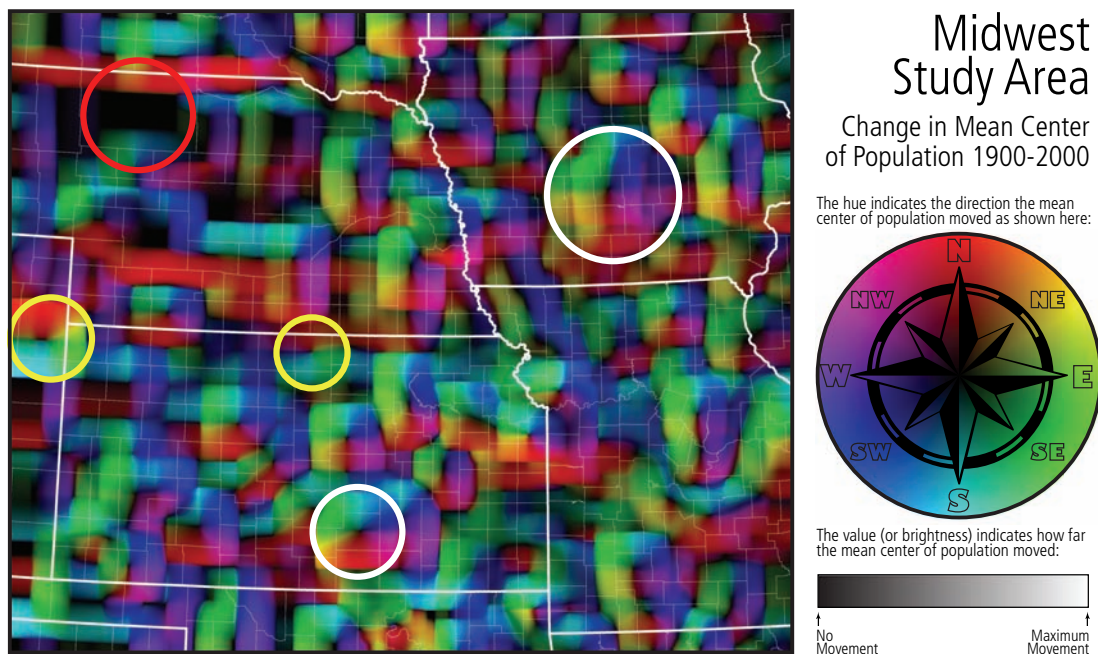


Figure 4.5. Central Midwest study area map with highlighted areas of interest.

On the other hand, the patterns in the yellow circles mimic to some extent the pattern in the legend compass rose. The circle near the center of the map highlights Jewell County, Kansas, which lost a considerable population from 1900 to 2000 (United States Bureau of the Census, 2007); over the course of the century, the mean centers of population moved away from the county. The circle on the left side of the map is an interesting case. This circle is centered over the southern part of Yuma County, Colorado. The reason for the pattern and the high magnitudes here is due to the restructuring of the counties between 1900 and 2000. In 1900, the southern half of Yuma County was part of the much larger Arapahoe County that extended west all the way to Denver—including most of the city (Figure 4.6)—and so in 1900 had a population of over 150,000. After the restructuring, the much smaller Yuma County had a population under 10,000 in 2000. This change in boundary structure accounts for the appearance of a large change in population, when in reality the population was merely administratively divided.

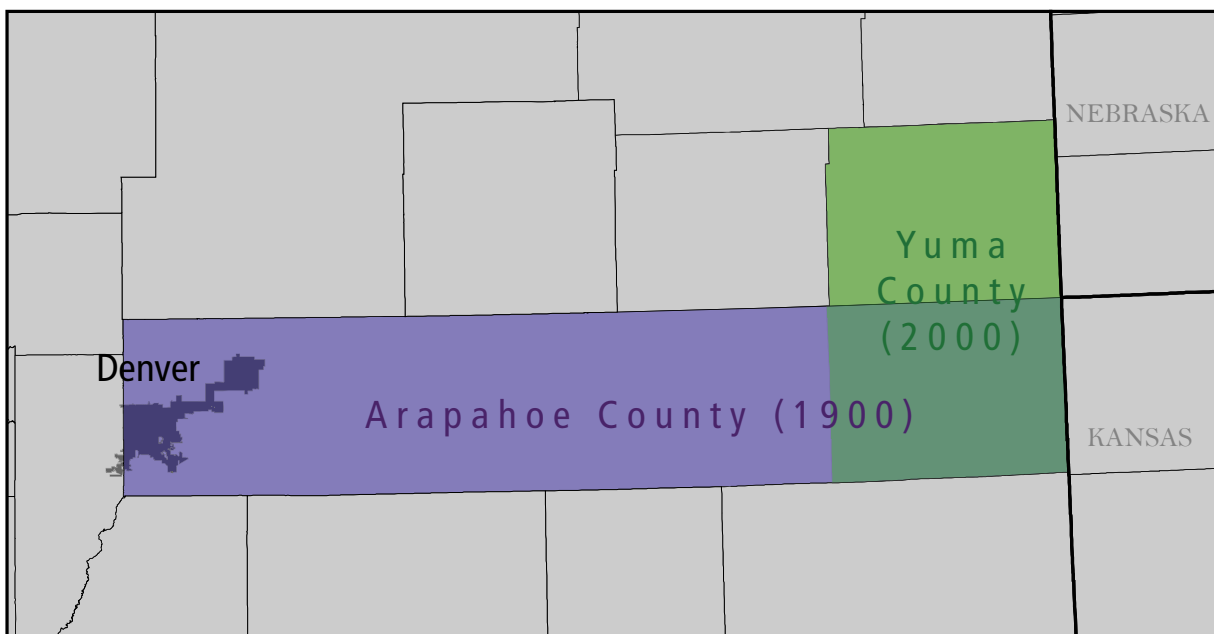


Figure 4.6. The old Arapahoe County from 1900 extending from the Kansas border all the way west to Denver, and the new restructured Yuma County in 2000.

The red circle in Figure 4.5 illustrates another issue; this circle is centered on Cherry County, Nebraska, which is the largest enumeration unit in the study area. In fact, it is large enough that our 55km moving window falls completely inside of it over a portion of its journey. Since the original census data that I used for these maps were aggregated at the county level, there is no information about the population distribution inside the county, and so it must be assumed to be uniform. As pointed out in Chapter 2, the mean center of population of a uniform distribution will always be the geographic center, and thus there will never be a change from year to year in these windows. The magnitudes in the inner portion of Cherry County will always be zero (thus, the black color), unless we increase the size of the moving window so that it will never be completely inside the boundaries of the county. The shortest dimension of the county is about 100km north to south, so a moving window of that size or greater will eliminate this issue.

Alternate Moving Window Sizes

I used the 55km moving window size for the sake of example in describing the process of generating the final map. However, we can use any moving window size we desire, simply by passing that parameter to the *cimc* function described at the end of Chapter 3. In the Central Midwest example, using smaller moving window sizes would increase the likelihood of running into the same issue we had with Cherry County, where the windows are smaller than the enumeration units. Because of this, I elected to try some larger sizes, namely 100km and 200km moving windows, the results of which are shown in Figures 4.7a and 4.7b, respectively.

The patterns in the maps created with the larger moving windows have much more defined patterns, but symbolize trends over a much larger area. In the 200km map, for example,

there are clear “donut” patterns converging on the aforementioned Des Moines and Wichita metropolitan areas, and also on Kansas City and Omaha.

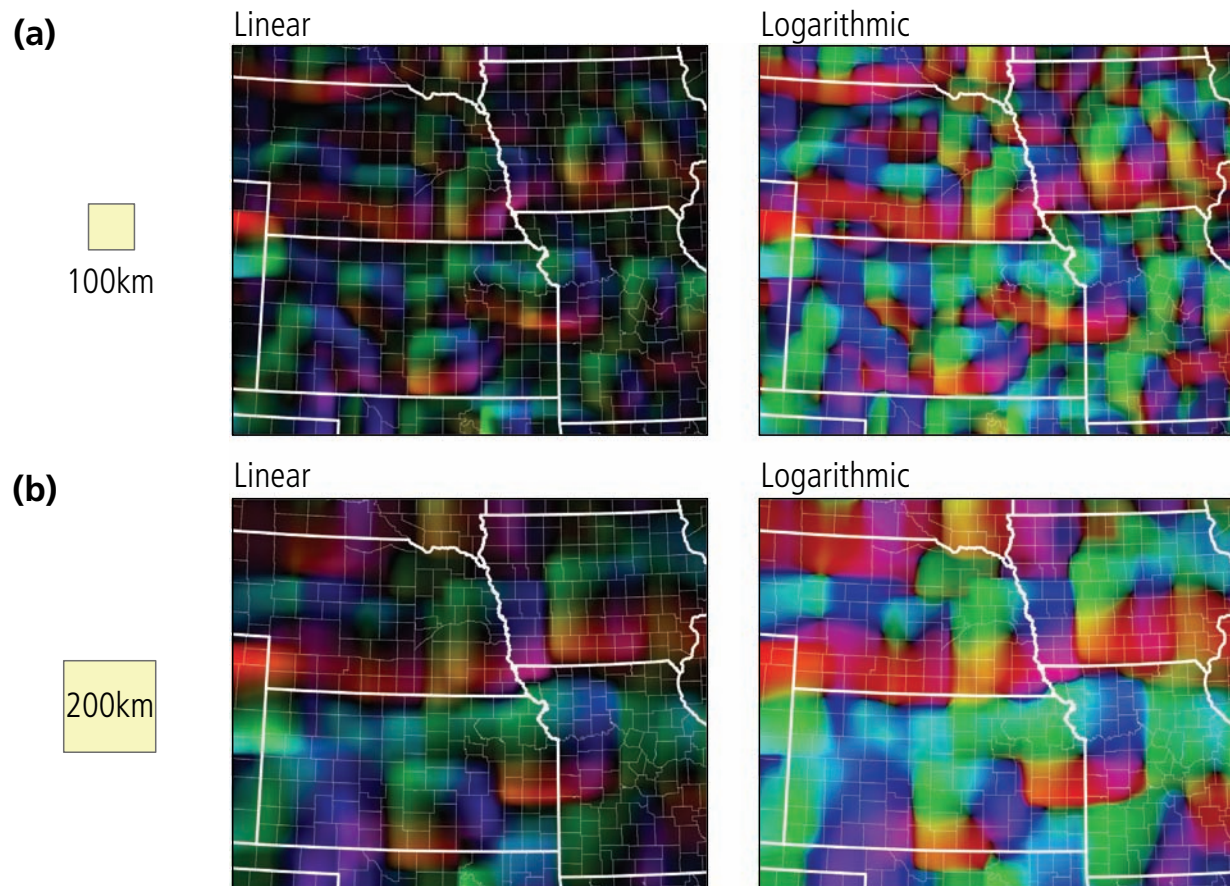


Figure 4.7. Linear and Logarithmic value scaling results for 100km and 200km moving windows in the Central Midwest study area.

The Jewell County pattern is much more prominent in the 200km map. Many of the smaller, more localized patterns evident in the 55km map are “drowned out” in the maps with the larger neighborhood sizes.

Alternate Study Area

After working for many months in the development of this process utilizing the same study area, upon completion I was understandably interested in seeing how the techniques would

apply to a new study area. Rather than simply choosing a different area at a similar scale, I decided to change the scale of both the study area and the input enumeration units. With the Central Midwest study area we were looking at an area 1,033 kilometers by 848 kilometers, and our population data were aggregated at the county level. Because of my familiarity with the area, I opted to move to a larger scale and repeat the process for the Kansas City metropolitan area. In this case, I gathered population data at the census block level, the smallest aggregate possible. This new study area covered an area of 68 kilometers x 57 kilometers, and I increased the resolution to 100 meters per cell to accommodate the smaller enumeration units.

The pre-processing for this study area was nearly the same as for the Central Midwest study area. Census block-level data for 1900 were not available, so I downloaded the data for 1990 and 2000, opting for a one-decade study rather than a one-century study. I joined the NHGIS data tables to the shapefiles in the same way as with the Central Midwest datasets, using the GISJOIN code. I made the determination that in this case the simplification of the datasets was not necessary; at larger scales the overall vertex counts are already much lower than at smaller scales. Next, I calculated the density field, this time calculating persons per 1/100th of a square kilometer. Since I opted to reduce the resolution to accommodate the smaller enumeration units of the census block dataset, I had 100 cells in each square kilometer. Finally, I clipped the datasets to the study area extent and converted them to the raster density grids for importation into MATLAB.

I ran these two density grids (1990 and 2000) through the *cimc* function using neighborhood sizes of 50, 100, and 200, with both linear and logarithmic scaling to yield six new maps. It is important to note that these neighborhood sizes refer to the number of cells, not a standard measurement. In the Central Midwest study area, the neighborhood size of 55

coincided with 55km windows because the cell size in that case was 1km x 1km. In this case, where the cell size is 100m x 100m, a neighborhood size of 55 would actually equate to a window size of 5,500 meters, or 5.5km. Figure 4.8 shows the six resultant maps (linear and logarithmic scaling with 5, 10, and 20 km moving windows) with county boundary and major highway overlays for context.

The patterns in the 5-kilometer moving window map reflect changes in suburban population distribution. The inverse “donuts”—those that show a convergence of population centers into a single area—show areas where new housing areas were built during the 1990 to 2000 time period. The patterns on this map show localized changes; on the other hand, the 20-kilometer moving window map shows a defined pattern encircling the entire city. In the areas north of the river, the mean centers of population have moved northward, away from the city center. In the east they have moved east and in the southwest on the Kansas side, they have moved to the southwest, while the mean centers of population near the city center have moved very little. This demonstrates the continuation of suburban sprawl that many large cities have been experiencing over the past century.

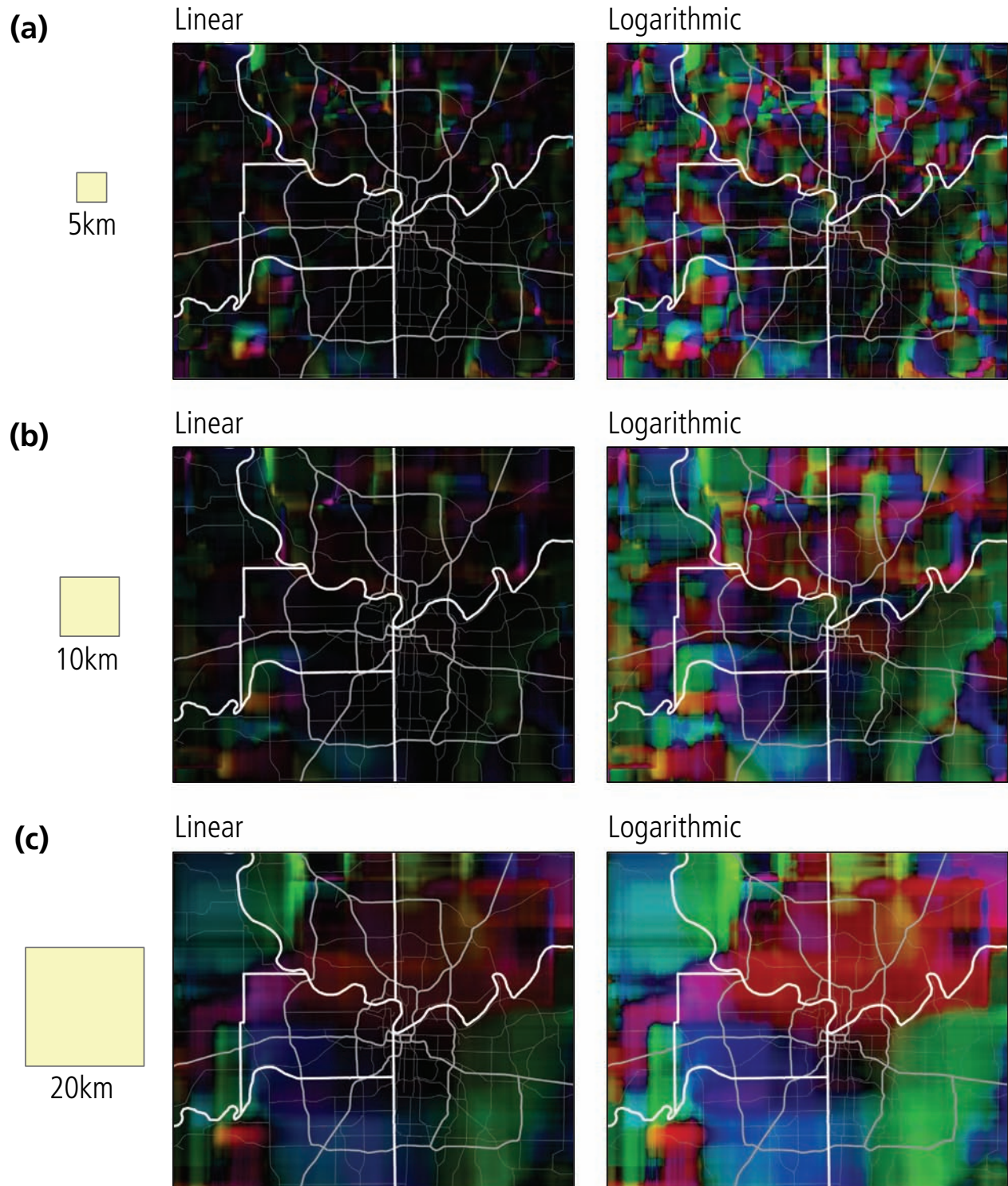


Figure 4.8. Linear and Logarithmic maps of the change in mean centers of population from 1990 to 2000 in the Kansas City study area using 5, 10 and 20 kilometer moving windows.

Assessing the Maps: Survey Development

Maps serve little purpose if they cannot be read and understood. Furthermore, ensuring that the map reader properly interprets data on the map is of paramount concern to every cartographer. In order to determine if people are able to read and interpret these maps, I designed a short survey.

Due to the complexity of the methods behind designing these maps, I initially thought it would be of value to provide some background into the cartographic process to the potential survey respondent before asking any questions. However, 12 pages and 14 figures into the background information writing process, I realized that it would be impractical to provide this level of information to everyone who would be reading these maps. In fact, I became concerned that with 12 pages of text to read, respondents would simply skip the reading and attempt to answer the questions anyway. With no way of knowing which respondents had actually read—or understood—the information, I decided to scrap all of it and simply present the maps with minimal information (as they would appear in a publication, for example) before asking the questions.

The survey I administered is shown in Appendix J. I utilized SurveyMonkey (www.surveymonkey.com), a popular site that allows users to design and administer surveys of many types. The site offers both free and paid plans, the latter with significantly enhanced administration tools as well as intuitive analysis tools. Limitations in what I was willing to spend on administering a single, small survey forced me to stick to the free plan, which gave me access to almost no analysis tools and limited the survey to ten questions. In practice, I was only able to ask eight questions, as two of the ten “free” questions were used up by the map images

themselves. To make things as painless as possible for the respondents, I made all but the final question multiple choice.

Page one of the survey had a standard introduction briefly describing the research and including much of the required information statement for human subject research. It also had a line of red text indicating that the respondent should resize the survey window until all the red text appeared on a single line. I did this to (hopefully) ensure that all the elements on the maps would be visible in the window. Page two presented the respondent with the finished 10km/logarithmic scaled Kansas City study area map with several areas outlined in yellow, along with questions 1 through 5. I geared the early questions to determine if the respondent could understand the map symbology by utilizing the legend. On page three, for questions 6 through 8, the respondent was presented with the 55km/logarithmic scaled Central Midwest study area map. The latter questions shifted more toward determining if the respondent was correctly interpreting the map, rather than simply matching values and hues to the legend. I will discuss the individual questions in more detail as I go over the results in the next section.

I deployed the survey first by having a colleague post a link on Facebook, as I was curious as to what type of response I would get by simply posting there. After one week I had 16 respondents, many of whom were likely college graduates in the social sciences. For phase two I reached out to some other colleagues directly in an attempt to have them administer the survey to their students, but this failed and I only racked up three additional respondents. Finally, I spent the money I was awarded for the Association of American Geographers Cartography Specialty Group Master's Thesis Grant on "targeted audience" responses from the SurveyMonkey site. Within a few hours, I had an additional 31 responses for a total of 50. Unlike the "free" responses, these included some demographic information about the respondents. Of the 29

respondents who volunteered demographic information (two did not), six (21%) were in the 18-29 age bracket, ten (34%) were in the 30-44 age bracket, 11 (38%) were in the 45-59 age bracket and the remaining two (7%) were 60 or over. Females accounted for nearly 76% of the respondents with 22 while the remaining seven were male.

The free version of the SurveyMonkey service does not allow for the exportation of the results into a useful format, so I had to manually enter the results for each respondent and each question into a spreadsheet for analysis.

Assessing the Maps: Survey Results

To examine the results of the survey, I will first discuss the results for each question individually, and then discuss the overall results as a whole.

Question 1 was asked principally to determine if the respondent was able to differentiate between low and high magnitudes as depicted on the map. The respondent was directed to four labeled areas on the map and asked to choose the area with the highest magnitude (Figure 4.9).



Figure 4.9. Question 1 of the survey. The green bar represents the correct answer.

While I attempted to choose areas on the map that would make the answer to this question as obvious as possible without compromising the intent of the question, there was still the possibility that the identification of the “brightest” area could be largely subjective. Ultimately, 66% of the respondents chose the “correct” answer, with the remaining respondents choosing the other three responses, at around to 10-12% each.

Question 2 was asked to determine if the respondent understood the meaning of the darker areas on the map (Figure 4.10). Two relatively dark areas on the map were highlighted, and the respondent was asked to choose the best answer describing what the map symbolized there. One important aspect of this question is the phrasing of the answers. Answer “b”, which says that the population stayed the same could be the correct answer to the question, but it is not the best answer; answer “c”, which states that the mean centers of population moved very little, is a more accurate description based on the map legend. Answer “c” was chosen by 64% of the respondents, while 22% of the respondents chose the somewhat correct but ultimately wrong answer of “b”. This question had a very high correlation with the overall scores; only one respondent who answered this question incorrectly scored better than 50% on the entire survey.

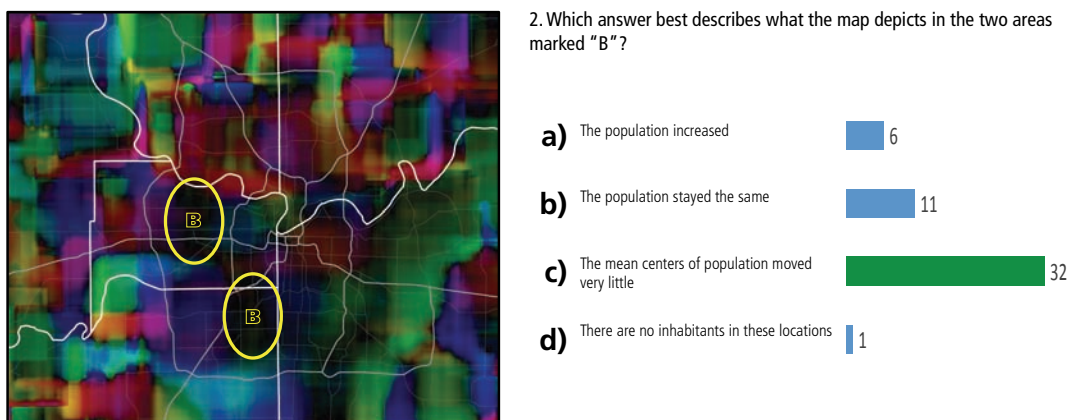


Figure 4.10. Question 2 of the survey. The green bar represents the correct answer.

Question 3 was asked to determine if the respondent understood the hue-direction relationship (Figure 4.11). A relatively monochromatic portion of the map was highlighted, and the respondent was given three possible directions to choose from. Of the respondents who scored 50% or better on the survey, only three missed this question. This question had the best correct response rate of all the questions in the survey with 80% of the respondents answering the question correctly.

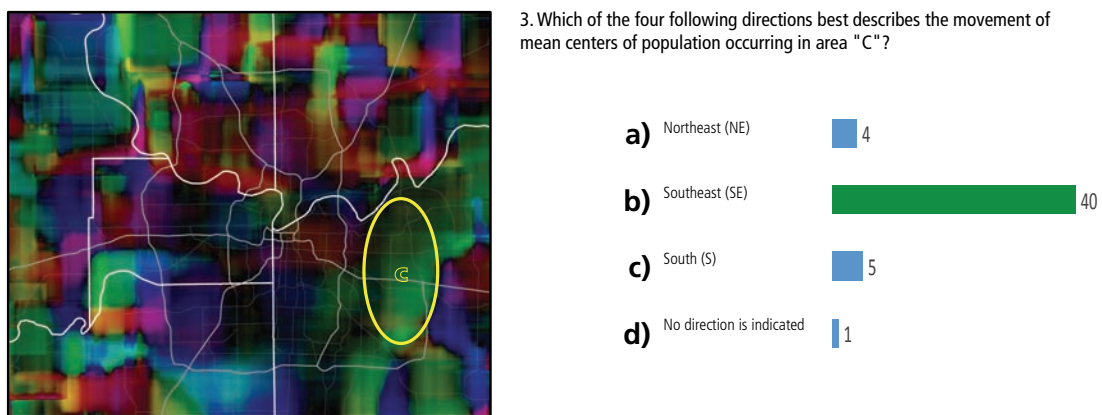


Figure 4.11. Question 3 of the survey. The green bar represents the correct answer.

Question 4 expands on question three by both showing a less obvious—but still noticeable—color pattern, and by incorporating some interpretation of the map symbolization into the question and possible answers (Figure 4.12). Thus, rather than just asking the respondent to determine the hue and refer back to the compass rose legend to discover the direction, the respondent was asked to answer what the symbology actually means. The correct answer was chosen by 60% of the respondents, with the other three answers being chosen by 12-14%.

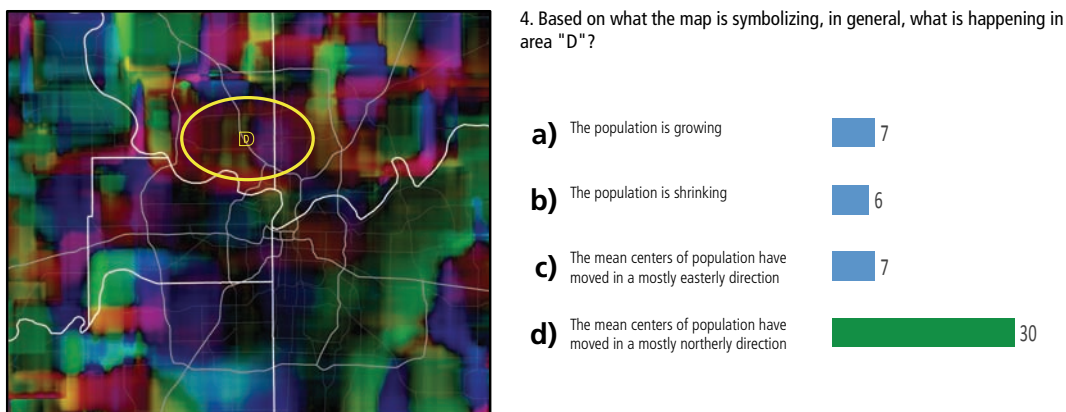


Figure 4.12. Question 4 of the survey. The green bar represents the correct answer.

Question 5 introduces the “donut” pattern, and simply asks the respondent what is depicted there (Figure 4.13). This question had the smallest percentage of correct answers with only 44%. This was not a surprise, as the concept of the convergence and divergence areas (the “donuts”) is one of the more difficult to understand.

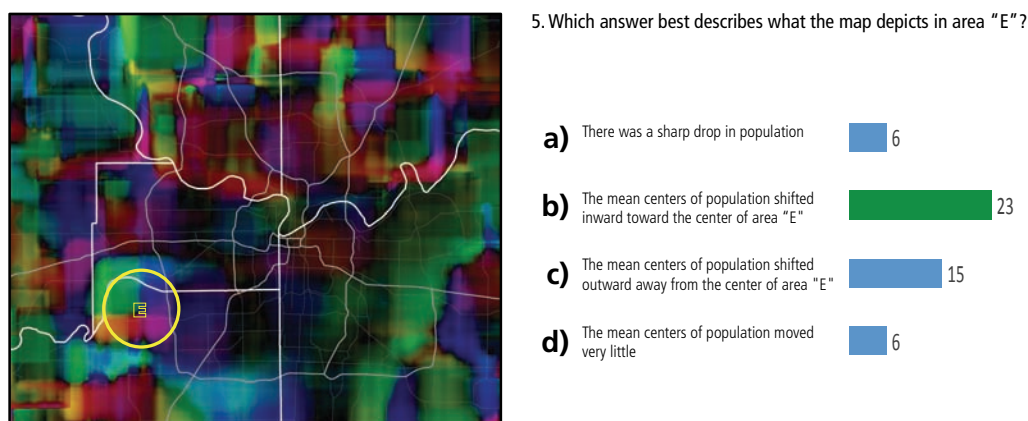


Figure 4.13. Question 5 of the survey. The green bar represents the correct answer.

Question 6 highlights two “donut” areas and gives the respondent some information about the study area (Figure 4.14). He or she is then asked to identify what type of area the “donut” area highlights. The respondents fared better here than on question 5, although this

question can be correctly answered without being able to correctly read the map. Anyone with familiarity of the study area would be able to deduce the correct answer to this question. Either way, 58% of the respondents chose the correct answer to this question.

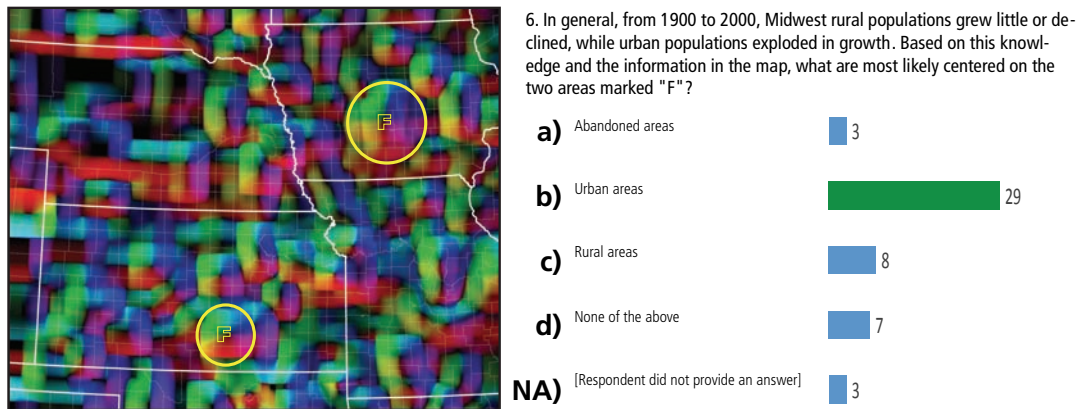


Figure 4.14. Question 6 of the survey. The green bar represents the correct answer.

Question 7 highlights an area where a major interstate highway runs through the area, and based on the information in the map, asks the respondent what a likely effect of the construction of the interstate had on the population dynamics (Figure 4.15). Again, a knowledge of the study area would allow the respondent to answer this question without the aid of the map, however, only 56% of the respondents chose the correct answer.

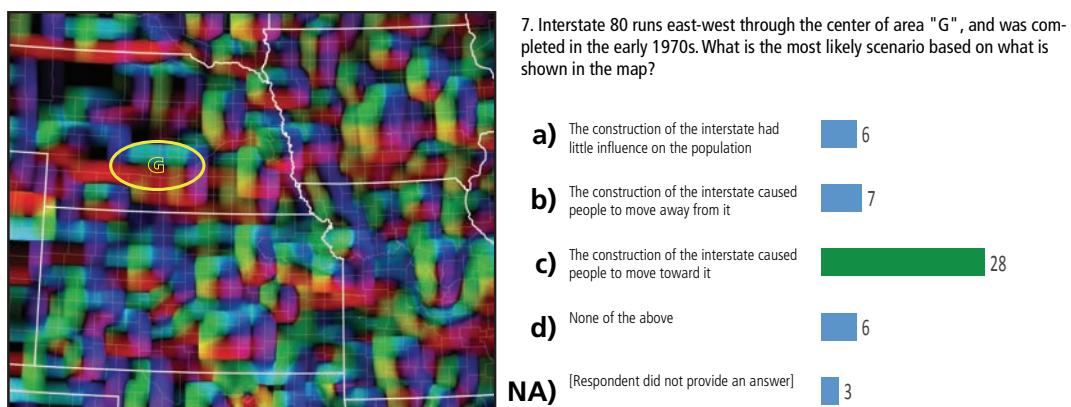


Figure 4.15. Question 7 of the survey. The green bar represents the correct answer.

Finally, question 8 essentially repeats what is asked in question 2, but leaves the answer open-ended (Figure 4.16). I did this primarily in an effort to gauge how seriously each respondent was taking the survey. The question requires at least a basic understanding of the map to answer accurately; respondents who did not read the instructions or make any attempt to understand the maps would fall flat in attempts to answer this question, even if they correctly guessed the answers on the multiple choice questions. Surprisingly, 70% of the respondents gave an answer that was more or less correct. I considered a correct answer to be any that had language to the effect of “no movement” or “little or no change”. 10% gave an answer that made it clear they had at least looked at the maps, but apparently did not understand them. Another 12% gave an answer that indicated they were completely clueless as to what was being asked or they were simply not taking the survey seriously. The remaining 8% provided no answers at all.

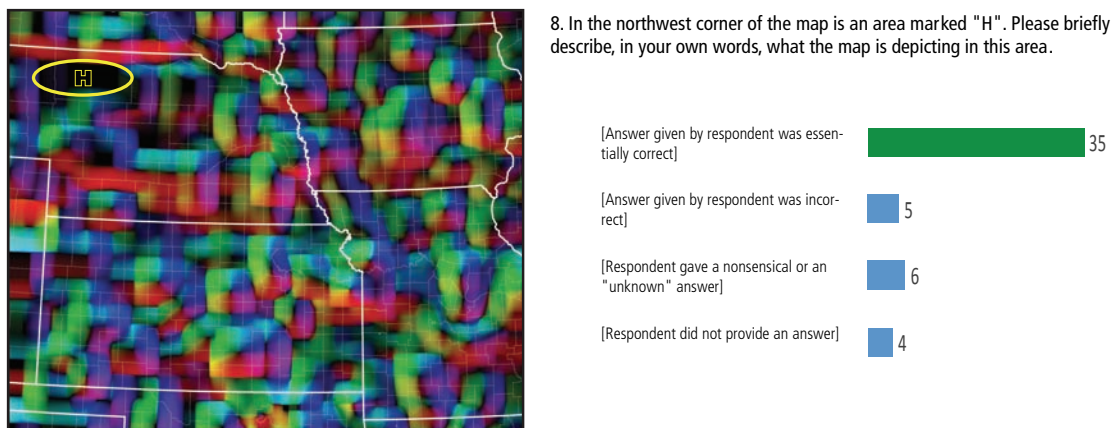


Figure 4.16. Question 8 of the survey. The green bar represents answers that were deemed correct.

Overall, 11 of the respondents, or 22% answered all the questions correctly. 10% missed one question, while 14% missed two. The average score on the survey was 5 correct out of 8, or 62.5%. Unfortunately, the average was dragged downward by respondents who were not making an effort to answer the questions. For example, two of the respondents chose “a” as the answer

for every question and provided nonsensical answers for question 8. Of the bottom nine scores (scores of 2, 1, and 0), it appeared that only two of the respondents were taking the survey seriously based on the answers to question 8 (see Appendix K). Another issue that may have brought down the mean score was that three of the respondents did not answer any of the questions on page three of the survey. Finally, there was no way for certain to tell if the respondents correctly resized their screens in order to see the entirety of the maps. If any of them failed to do this, it would certainly reflect on their scores, as critical components such as the map legends would have been obscured from view.

Figure 4.17 shows the results of the overall survey. The green bars represent the respondents who seemed to be taking the survey seriously based on their answer to question 8. The orange bars represent those respondents who did not reply at all to question 8, indicated that they did not know the answer, or responded with nonsense. The results exhibit some negative skew, which seems to indicate that all-in-all, the maps were understood by more respondents than did not understand them.

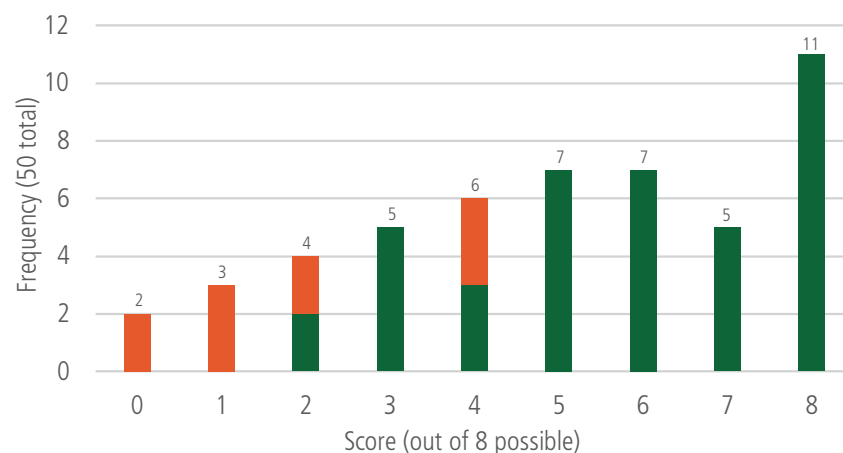


Figure 4.17. Overall results of the survey. The orange portions of the bars represent the results from respondents that were categorized as not taking the survey seriously.

Despite the positive results, one concern I had is the possibility of some bias in the sampling, as the first round of recruiting was done through the social media networks of my colleagues, while the second paid round was done through the SurveyMonkey “target audience” service. Because of this, it seemed there may have been a good chance that those who responded from the first round had at least some college education. However, after examining the responses from both the first and second groups, it turned out that there were good and poor results from both groups. Of the 11 perfect scores, five of them came from respondents in the paid “target audience” pool, while the other six were from the recruited group. Of the scores of 4 and lower, six were recruited and 13 were paid. Ultimately, I have no method of measuring whether or not there was any bias in the sampling, as no personal data were collected.

In Chapter 5, I will discuss some conclusions I have made during the writing of this thesis as well as some of the things I wish I could have done, in the event I or anyone else ever decides to continue with this research in the future.

Chapter 5

Conclusions

In Chapter 1, I introduced the notion of mapping the changes in mean centers of population in moving windows, based in part on the much simpler precedent of mapping the mean centers of population for the United States done by the Bureau of the Census. In Chapter 2, I discussed the process of generating a vector field based on these changes in moving windows, and in Chapter 3 I discussed a method for symbolizing the vector field using the HSV color model. Finally, in Chapter 4, I discussed creating the finished map products, a few of the issues with the maps, and the assessment of their legibility based on responses to a survey.

I went into this project with a simple curiosity of what the finished map of the changes of mean centers in moving windows would look like. When I arrived at the initial direction-hue result (Figure 5.1), I didn't know what to make of it.

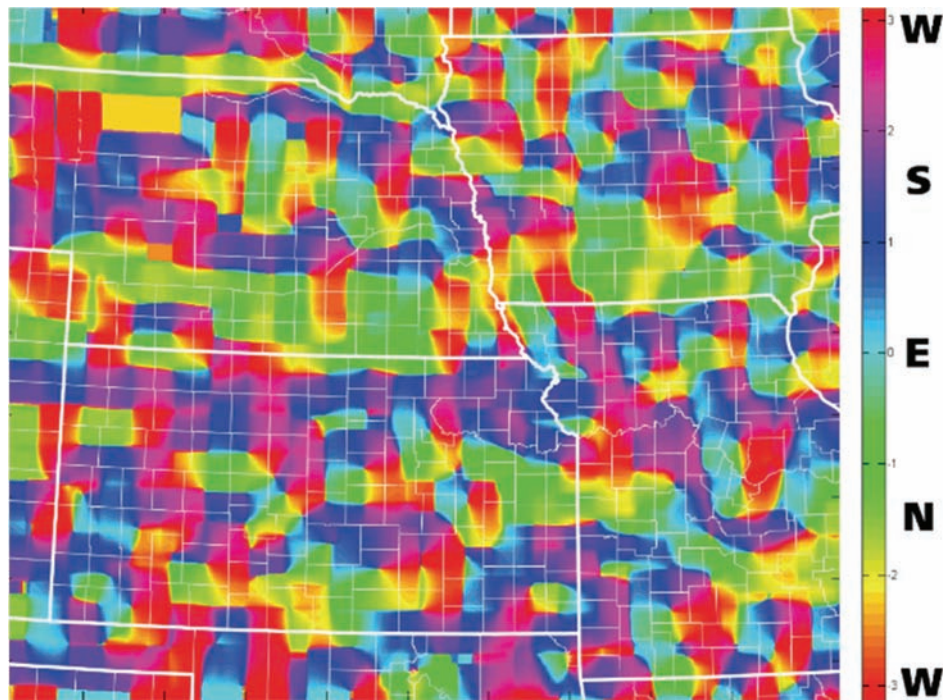


Figure 5.1. Initial hue-direction result map.

I shared it with colleagues, and while most of them noted that it looked “interesting”, I didn’t walk away feeling as though I had done anything remarkable. I found myself staring at the map, trying to make some sense out of it, and aside from some of the more obvious patterns that I have discussed earlier, I was unable to glean anything meaningful from it.

At first glance, it is difficult to discern what these maps tell you that cannot be communicated with more traditional maps. The same basic patterns that I’ve discussed in these maps are also evident in the choropleth maps of the study areas. The urban growth in the Central Midwest study area symbolized by the “donuts” in the change in mean center maps is clearly visible in the choropleth map, symbolized with the dark shades of blue (Figure 5.2). Each of the “donuts” highlighted in the map on the left corresponds to a large increase in population as shown in the choropleth map on the right. However, there are subtle differences in the color patterns that comprise the “donuts” that offer clues highlighting some of the differences in the dynamics of the population change. The colors indicate directional shifts in population centers of mass that are not evident in the choropleth map.

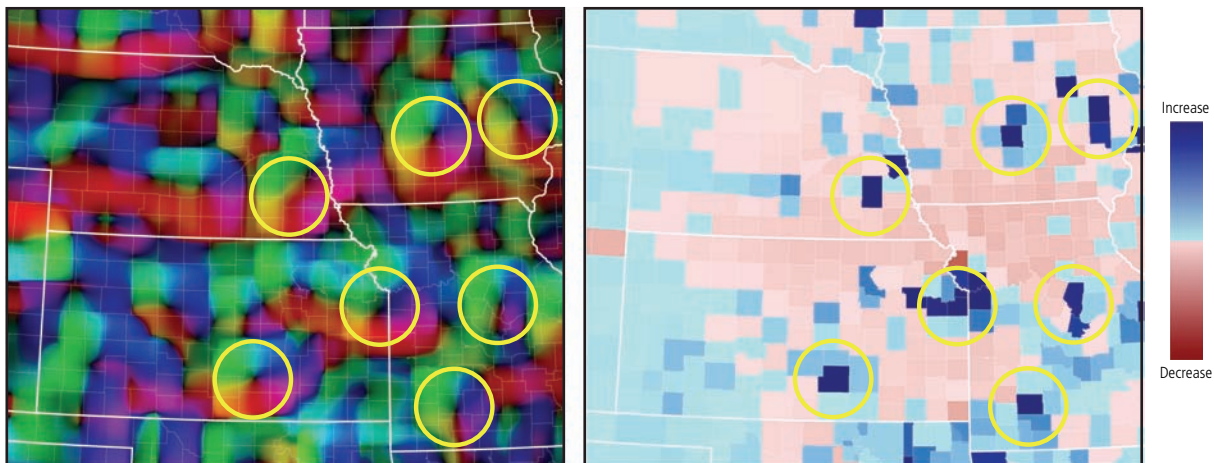


Figure 5.2. Change in mean center and choropleth map of the change in population from 1900 to 2000 at the county level for the Central Midwest study area. The yellow circles highlight urban growth areas.

Similarly, the suburban sprawl indicated in the Kansas City change in mean center map is visible in its choropleth counterpart as well, although perhaps not quite as obviously. The higher prevalence of the darker blue census blocks on the periphery of the city outline this fairly well, and correspond somewhat to the unidirectional hues in the change in mean center map on the left in Figure 5.3.

Despite the dubious utility of these maps, I do believe that an advantage in this mapping technique is the ability to look at the data in a number of different ways, based on the sizes of the moving windows that are chosen. The smaller windows generate a more localized “story” while the large ones give us the sweeping trends over large areas.

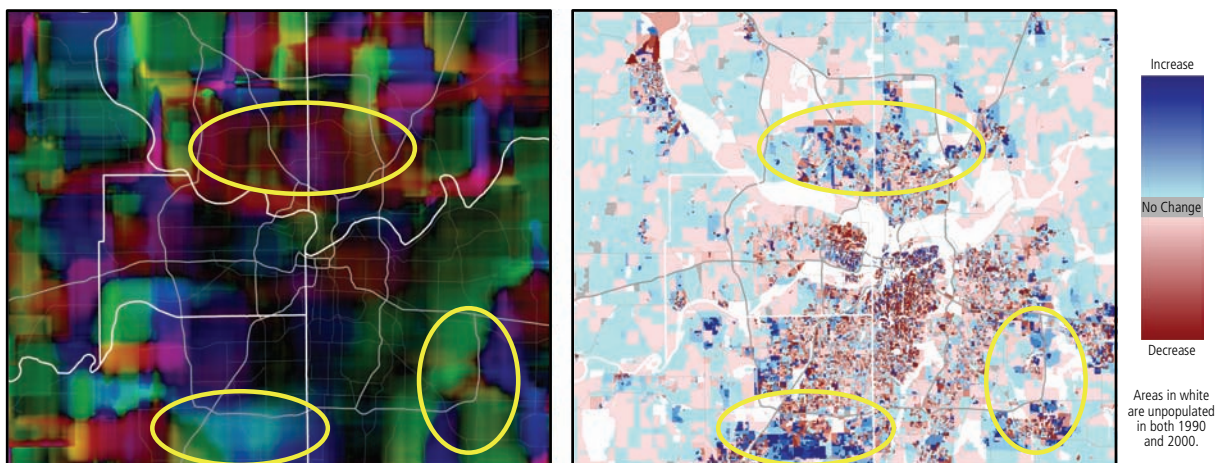


Figure 5.3. Change in mean center and choropleth map of the change in population from 1990 to 2000 at the census block level for the Kansas City study area. The yellow circles highlight areas of significant increases of suburban population.

Further Research

First, I feel as if a more in-depth color-direction association survey might shed some light on the responses that I received in the one that I administered. Determining the reasons—if

any—for the respondents’ color-direction choices could be instrumental in the development of a more intuitive symbolization scheme.

It would also have been interesting to experiment with different scales of data, that is, use census block data and county-level data for the same study area. If I had access to 1900 census data at the block level, this would have been possible; however, I had to make do with the 1990 data. Exploring the data at finer geographical levels would help reveal the limitations introduced by using data aggregated at geographically larger enumeration units.

Another interesting possibility would be to map the change in the change, that is, we could utilize population data at regular intervals and plot the change vectors for each interval (e.g., 1900-1910, 1910-1920, etc.). We could then conceivably map the changes in direction and magnitude of the vectors from period to period—a sort of “acceleration/deceleration” map.

I have also pondered the idea of utilizing the saturation component of the HSV model to strengthen the visualization of the magnitudes. Perhaps adjusting this component in tandem with the value component rather than leaving it static throughout would produce sharper results.

Finally, I feel as if there could be a more in-depth assessment of the maps. Perhaps the utilization of focus groups comprised of cartographically inclined individuals could help with understanding some of the patterns.

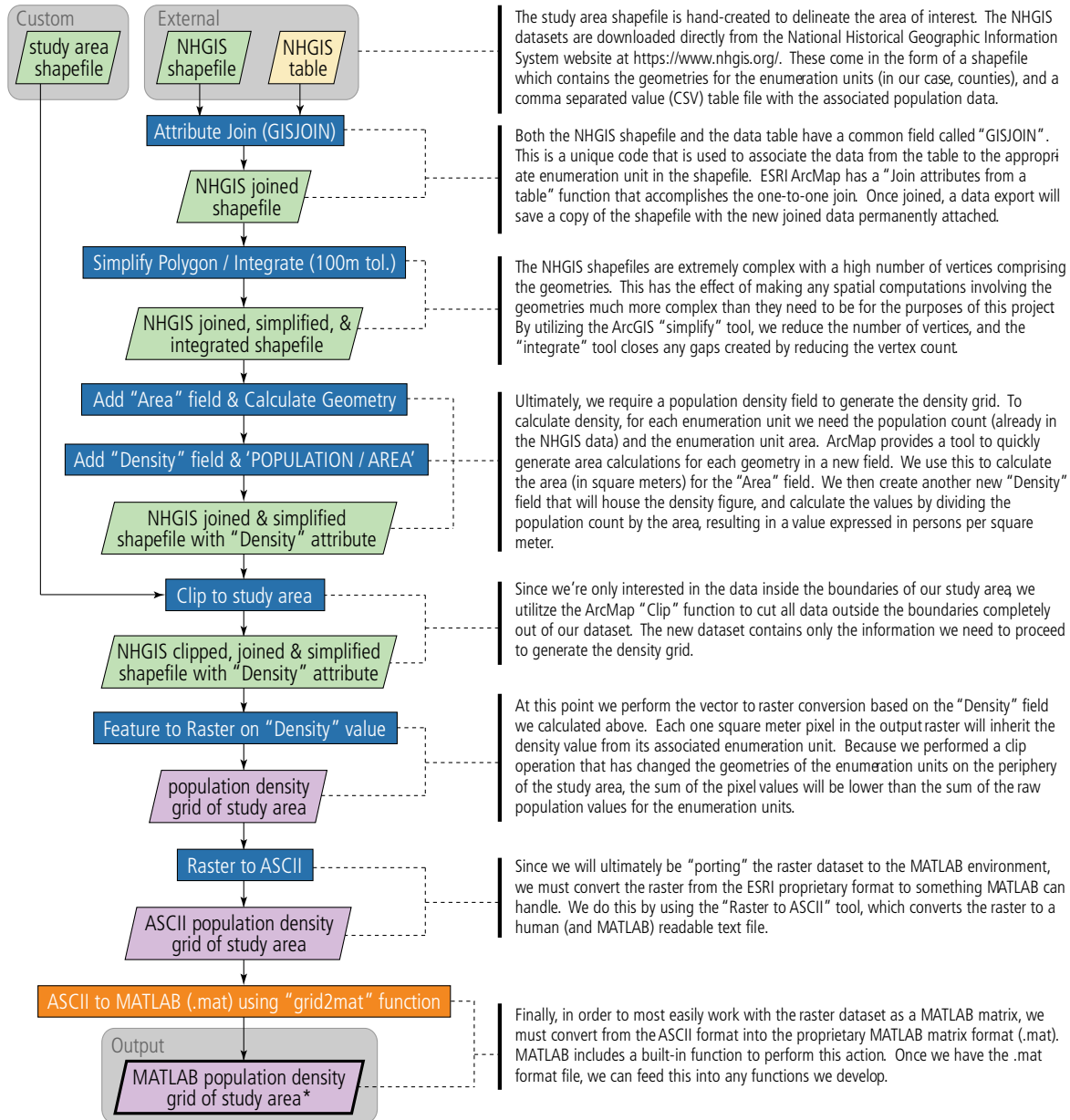
References

- Aisch, G., & Gebeloff, R. (2014). Mapping Migration in the United States. *The Upshot, The New York Times*. Retrieved from: <http://www.nytimes.com/2014/08/16/upshot/mapping-migration-in-the-united-states-since-1900.html>
- Balzer, M., & Deussen, O. (2005). Voronoi Treemaps. *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*. 49-56, Washington DC, IEEE Computer Society.
- Barmore, F. E. (1993). Where are we? Comments on the Concept of "Center of Population." *Solstice: An Electronic Journal of Geography and Mathematics*, 3(2)
- Boring, E., & Pang, A. (1996). Directional Flow Visualization of Vector Fields. Visualization '96 Proceedings. Retrieved September 14, 2008, from <ftp://ftp.cse.ucsc.edu/pub/reinas/papers/vis96.directional.ps.gz>
- Brewer, C. A., & Marlow, K. A. (1993). Color Representation of Aspect and Slope Simultaneously. Proceedings, Eleventh International Symposium on Computer-Assisted Cartography, Oct/Nov, pp. 328-337. Minneapolis.
- Eastman, R. J. (1986). Opponent Process Theory and Syntax for Qualitative Relationships in Quantitative Series. *The American Cartographer*, 13(4), 324-333.
- Greenfield, J. (2013, February 7). Students Still Not Taking to E-Textbooks, New Data Show. Retrieved April 13, 2015, from <http://www.digitalbookworld.com/2013/students-still-not-taking-to-e-textbooks-new-data-show/>
- Image Analyst (2014). How to make a colorwheel in MATLAB [Answer 2, 23 September, 2014]. Posted to: <http://www.mathworks.com/matlabcentral/answers/155877-how-to-make-a-colorwheel-in-matlab>.
- Li, X., & Hodgson, M. E. (2004). Vector field data model and operations. *GIScience and Remote Sensing*, 41(1), 1-24.
- Minnesota Population Center. (2011). National Historical Geographic Information System: Version 2.0. Minneapolis, MN: University of Minnesota 2011. Retrieved from <http://www.nhgis.org>
- Moellering, H., & Kimerling, A. J. (1990). A New Digital Slope-Aspect Display Process. *Cartography and Geographic Information Systems*, 17(2), 151-159.
- Newspaper Association of America. (2015). Newspaper Circulation Volume. Retrieved April 13, 2015, from <http://www.naa.org/Trends-and-Numbers/Circulation-Volume/Newspaper-Circulation-Volume.aspx>

- Pew Research Center. (2015). News Magazines: Overall Circulation by Publication. Retrieved April 13, 2015, from <http://www.journalism.org/media-indicators/news-magazine-overall-circulation/>
- Slocum, T. A., McMaster, R. B., Kessler, F. C., & Howard, H. H. (2009). *Thematic Cartography and Geovisualization*. Upper Saddle River, NJ: Pearson Prentice Hall.
- Tobler, W. R. (1987). Experiments in migration mapping by computer. *The American Cartographer*, 14(2), 155-163.
- Tomlin, C. D. (1990). Geographic information systems and cartographic modeling. Englewood Cliffs, NJ: Prentice Hall.
- Travis, D. (1991). Effective Color Displays - Theory and Practice. San Diego, CA: Academic Press.
- United States Bureau of the Census. (2007). Retrieved from <http://www.census.gov/>
- Waterman, T. T. (1910). Religious Practices of the Diegueño Indians. University of California Publications in American Archaeology and Ethnology, 8(6), 271-358.

Appendices

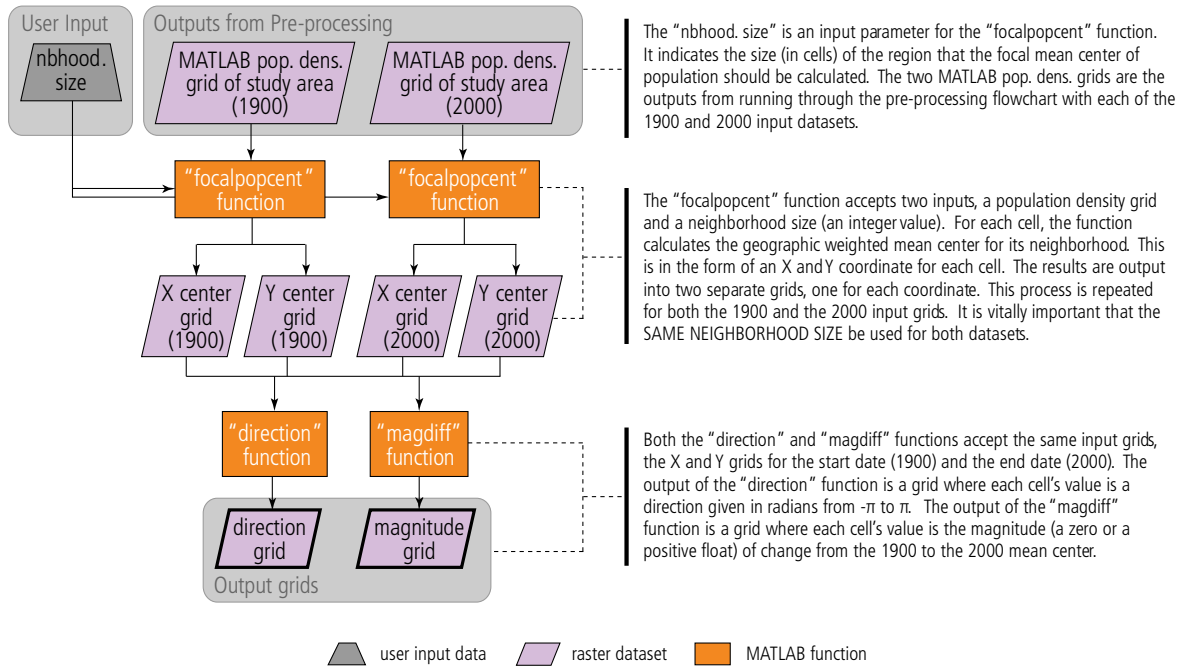
Appendix A. Pre-processing Flowchart



***NOTE: This entire process must be performed for BOTH sets of input datasets (the 1900 and the 2000) downloaded from NHGIS, resulting in two output grids.**



Appendix B. Direction and Magnitude Calculation Flowchart



Appendix C. Code for Color Survey

```

/*****
Name:          Color_Survey - Color - direction survey

Type:          Shockwave Flash ActionScript

Author:        Keith French

Synopsis:       This code will present the respondent with eight color "nodes" and
                 eight "receptacles" in a compass rose and ask the respondent to drag
                 and drop those nodes into the receptacles as they see fit. The code
                 is fairly robust in that it requires exactly one node in each
                 receptacle and makes it easy for the user to place them there. The
                 results are then transmitted to a php script on a server that records
                 them in a file.

Syntax:        N/A

Inputs:        N/A

Output:        Node/Receptacle association results, colorblind bit, and confirmation
                 code.

Objects:        confNum - The confirmation number
                 finish_bt - Finish button
                 reset_bt - Reset button
                 node[X]_mc - Movable nodes (8 total, [X] replaced by index)
                 start[X]_mc - Node starting placeholders (8 total, [X] replaced by index)
                 slot[X]_mc - Compass rose slots (8 total, [X] replaced by index)

*****/
// MAIN PROGRAM

// Variable declarations
var thresh:int=25;
var sp:Array = new Array(1,2,3,4,5,6,7,8);
var plg:Array = new Array(0,0,0,0,0,0,0,0);
var done:Array = new Array(0,0,0,0,0,0,0,0);
var nodes:Array = new
Array("node1_mc","node2_mc","node3_mc","node4_mc","node5_mc","node6_mc","node7_mc","node8_mc");

// Hide the confirmation number until the end and starts the movie
confNum.visible = false;
gotoAndStop(1);

// Disable and gray out the "finish button"
finish_bt.enabled = false;
finish_bt.alpha = .3;

// Associate the button objects with their functions
reset_bt.addEventListener(MouseEvent.CLICK, eReset);
finish_bt.addEventListener(MouseEvent.CLICK, saveData);

// Initialize the interface
randomize(sp);
reset(sp);

// Initialize the nodes
for (i = 0; i < nodes.length; i++) {
    var node:MovieClip = getChildByName(nodes[i]) as MovieClip;
    node.buttonMode = true;
    node.addEventListener(MouseEvent.MOUSE_DOWN, nodePickUp);
    node.addEventListener(MouseEvent.MOUSE_UP, nodePutDown);
}

//-----
// RANDOMIZE - This function randomizes the order of the elements of the input array.
```

```

function randomize(a:Array):void {
    var i:Number = a.length;
    if (i > 0) {
        while (--i) {
            var j:Number = Math.floor(Math.random() * (i + 1));
            var tmpi:Object = a[i];
            var tmpj:Object = a[j];
            a[i] = tmpj;
            a[j] = tmpi;
        }
    }
}

//-----
// RESET - This function returns the node objects to their original starting positions,
//         resets all previously set node placement values to zero, and ensures the
//         finish button is disabled.

function reset(a:Array):void {
    var i:int;
    for (i = 1; i <= a.length; i++) {
        this["node"+i+"_mc"].x = this["start"+a[i-1]+"_mc"].x;
        this["node"+i+"_mc"].y = this["start"+a[i-1]+"_mc"].y;
        plg[i] = 0;
        done[i] = 0;
    }
    finish_bt.enabled = false;
    finish_bt.alpha = .3;
}

//-----
// ERESET - Calls the "reset" function for the "sp" array.

function eReset(event:MouseEvent):void {
    reset(sp);
}

//-----
// NODEPICKUP - Code to handle the event when the user clicks and drags a node.
//             Since dragging a node implies it is to be removed from its previously
//             placed position, the previous position for the node is zeroed out.

function nodePickUp(event:MouseEvent):void {
    event.target.startDrag(true);
    event.target.parent.addChild(event.target);
    var node:int = -1;
    for (i = 1; i <= 8; i++) {
        if(event.target==this["node"+i+"_mc"]) node = i;
    }
    plg[done[node]] = 0;
    done[node] = 0;
}

//-----
// NODEPUTDOWN - Code to handle the event when the user drags a node to a location
//              and lets go of the mouse button (drops the node). This checks for
//              whether or not the node is close enough to a receptacle to "snap" into it
//              based on the threshold value, and whether or not the receptacle is not
//              already occupied by a node. If both are true, the receptacle is assigned
//              the value of the node, and the "clickSound" function is called. If there
//              is a successful placement of a node, the function calls the "checkNodes"
//              function which checks to see if all the nodes have been successfully
//              placed, and if so, it calls the "doneSound" function and enables the
//              "finish" button.

function nodePutDown(event:MouseEvent):void {
    event.target.stopDrag()
    var node:int = -1;
    for (i = 1; i <= 8; i++) {
        if(event.target==this["node"+i+"_mc"]) node = i;
    }
}

```

```

var curX:int = this["node"+node+"_mc"].x;
var curY:int = this["node"+node+"_mc"].y;
for(i = 1; i <= 8; i++) {
    if (curX > this["slot"+i+"_mc"].x - thresh &&
        curX < this["slot"+i+"_mc"].x + thresh &&
        curY > this["slot"+i+"_mc"].y - thresh &&
        curY < this["slot"+i+"_mc"].y + thresh &&
        plg[i] == 0) {
        plg[i] = node;
        done[node] = i;
        clickSound();
        this["node"+node+"_mc"].x = this["slot"+i+"_mc"].x;
        this["node"+node+"_mc"].y = this["slot"+i+"_mc"].y;
    }
    if (checkNodes()) {
        doneSound();
        finish_bt.enabled = true;
        finish_bt.alpha = 1;
    }
    else {
        finish_bt.enabled = false;
        finish_bt.alpha = .3;
    }
}

}

//-----
// CLICKSOUND - Plays a clicking sound.

function clickSound():void {
    var req:URLRequest = new URLRequest("click.mp3");
    var s:Sound = new Sound(req);
    s.play();
}

//-----
// DONESOUND - Plays a beeping sound.

function doneSound():void {
    var req:URLRequest = new URLRequest("done.mp3");
    var s:Sound = new Sound(req);
    s.play();
}

//-----
// CHECKNODES - This function checks to see if all the nodes have been successfully
// placed into the receptacles.

function checkNodes():Boolean {
    var finished:Boolean;
    finished = true;
    for(var j:int = 1; j <= 8; j++) {
        if (plg[j] < 1) finished = false;
    }
    return finished;
}

//-----
// DATAONLOAD - Dummy function used during debugging to test if data writing was
// successful.

function dataOnLoad(event:Event):void {
    //if(event.target.data.writing=="Ok") {
        gotoAndStop(2);
    //}
    //else trace("No 'OK' from php script!");
}

//-----
// SAVEDATA - Function that is called when "finish" button is clicked. This compiles
// all the node data, the colorblind bit, and generates the random

```

```

//          confirmation number, then sends the text string of the results to the
//          php script on the server to store the data.

function saveData(event:Event):void {
    if(finish_bt.enabled) {
        var blindChk:int = 0;
        if(blind.selected) {blindChk = 1;}
        var confirm:Number = Math.floor(Math.random() * 100000);
        confNum.text = String(confirm);
        var sendData:String =
            "compass="+plg.join(",")+","+blindChk+","+confNum.text;
        var variables:URLVariables = new URLVariables();
        variables.compass = sendData;
        var dataFile:URLRequest = new URLRequest();
        dataFile.url = "http://people.ku.edu/~kaf/cgi-bin/savedata.php";
        dataFile.method = URLRequestMethod.POST;
        dataFile.data = variables;
        var loader:URLLoader = new URLLoader();
        loader.dataFormat = URLLoaderDataFormat.TEXT;
        loader.addEventListener(Event.COMPLETE, dataOnLoad);
        loader.load(dataFile);
    }
}

```

Appendix D. The php Script to Store the Survey Results.

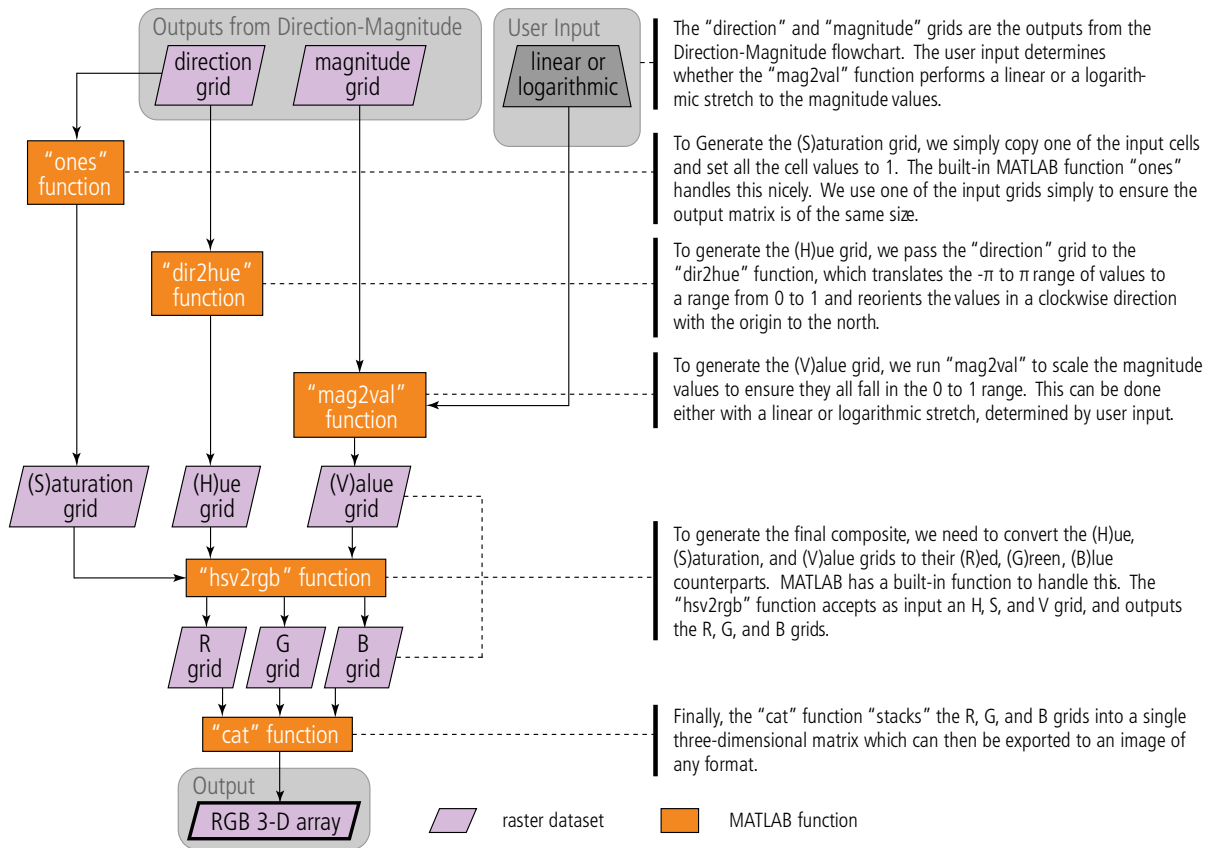
```
#!/usr/local/bin/php
<?php
$data = $_POST['compass'];
$outp = "$data\n";
$fp = fopen("outdata.txt", "a+");
if(fwrite($fp, $outp)) echo "writing=Ok&";
else echo "writing=Error&";
fclose($fp);
?>
```

Appendix E. Color Survey Raw Results

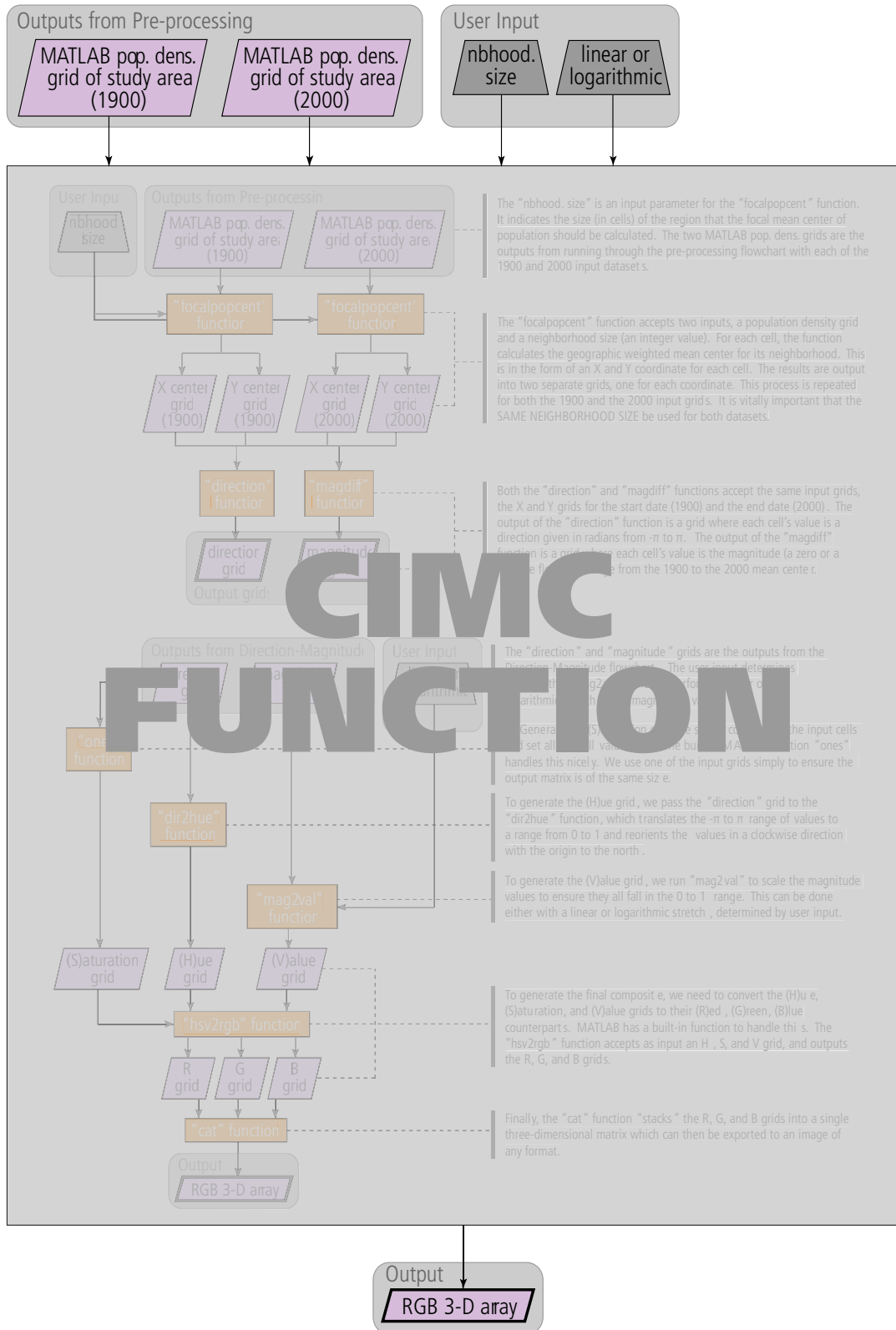
compass=0,6,7,5,4,3,2,1,8,0,1647
compass=0,6,7,3,2,1,8,5,4,0,69369
compass=0,1,2,3,4,5,7,6,8,0,91514
compass=0,3,2,1,8,6,7,5,4,0,33775
compass=0,1,2,3,4,5,7,6,8,0,1346
compass=0,1,2,3,4,5,7,6,8,0,9271
compass=0,6,7,3,2,1,8,5,4,0,38289
compass=0,5,4,3,2,1,8,6,7,0,94924
compass=0,3,2,4,5,6,7,1,8,0,16432
compass=0,1,8,5,4,3,2,7,6,0,83812
compass=0,1,8,6,7,5,4,3,2,0,59970
compass=0,1,2,3,7,6,5,4,8,0,92701
compass=0,7,4,3,8,2,1,6,5,0,56749
compass=0,7,4,3,8,2,1,6,5,0,26953
compass=0,6,7,8,1,2,3,5,4,0,48059
compass=0,1,2,3,4,5,7,6,8,0,16071
compass=0,5,6,8,7,1,2,3,4,0,85048
compass=0,1,8,6,7,5,4,2,3,0,12120
compass=0,1,8,2,3,7,4,5,6,0,80165
compass=0,2,3,5,4,7,1,6,8,0,43482
compass=0,1,8,6,7,5,4,3,2,0,50668
compass=0,1,8,6,7,5,4,3,2,0,55509
compass=0,6,7,5,4,3,2,1,8,0,59792
compass=0,1,2,3,4,5,6,7,8,0,71624
compass=0,6,7,5,4,3,2,1,8,0,49371
compass=0,4,8,3,5,7,6,2,1,0,94777
compass=0,7,6,5,4,3,2,1,8,0,53933
compass=0,3,2,8,1,4,5,7,6,0,83217
compass=0,1,2,3,4,5,7,6,8,0,79263
compass=0,5,6,2,1,4,8,7,3,0,30707
compass=0,6,5,7,4,2,3,1,8,0,16718
compass=0,1,2,7,4,8,3,6,5,0,11913
compass=0,5,7,6,8,1,2,3,4,0,24465
compass=0,1,2,3,4,5,7,6,8,0,51408
compass=0,1,2,3,4,5,7,6,8,0,38884
compass=0,1,3,2,4,5,7,6,8,0,21730
compass=0,5,4,6,8,1,3,2,7,0,5512
compass=0,1,2,3,4,5,6,7,8,0,26961
compass=0,1,2,3,4,5,7,6,8,0,97097
compass=0,1,2,3,4,5,7,6,8,0,45974
compass=0,6,7,8,2,1,3,5,4,0,5647
compass=0,1,2,3,4,5,7,6,8,0,5261
compass=0,1,2,3,4,5,7,6,8,0,21909
compass=0,1,3,6,2,4,7,5,8,0,89647
compass=0,6,4,3,7,5,2,1,8,0,7319
compass=0,6,7,3,4,5,2,1,8,0,85592
compass=0,6,4,5,8,1,3,2,7,0,76756
compass=0,5,6,4,7,3,2,1,8,0,62819
compass=0,1,2,3,4,5,6,7,8,0,37918
compass=0,1,2,3,7,6,4,5,8,0,93412
compass=0,6,8,3,4,1,5,2,7,0,99477
compass=0,1,7,6,4,5,3,2,8,0,98846
compass=0,1,6,7,5,4,3,2,8,0,92507
compass=0,1,6,7,8,3,2,5,4,0,53990
compass=0,1,8,6,7,4,5,3,2,0,39253
compass=0,6,4,5,8,1,3,2,7,0,6165
compass=0,1,3,2,6,7,4,5,8,0,57404
compass=0,4,5,7,2,8,6,3,1,0,49160
compass=0,1,3,2,4,5,7,6,8,0,82118
compass=0,6,7,1,8,2,3,4,5,0,35964
compass=0,6,7,4,5,3,2,1,8,0,67021
compass=0,1,2,3,4,5,7,6,8,0,54395
compass=0,6,4,5,8,1,3,2,7,0,80668
compass=0,1,2,3,4,5,6,7,8,0,20582
compass=0,6,7,5,4,2,3,1,8,0,74526
compass=0,4,2,3,1,8,7,6,5,0,91799
compass=0,1,2,3,4,5,7,6,8,0,24307
compass=0,1,8,6,7,5,4,3,2,0,4811
compass=0,1,2,3,4,5,7,6,8,0,20098
compass=0,1,2,3,4,5,7,6,8,0,31954
compass=0,1,3,2,5,4,6,7,8,0,94635

compass=0,3,7,4,5,2,1,6,8,0,43762
compass=0,6,2,1,5,3,7,4,8,1,28969
compass=0,4,5,7,6,8,1,2,3,0,65658
compass=0,4,5,6,8,1,2,3,7,0,35078
compass=0,1,2,3,4,5,6,7,8,0,13512
compass=0,1,2,3,4,5,6,7,8,0,67238
compass=0,5,6,7,8,1,2,3,4,0,48816
compass=0,1,5,4,7,6,8,3,2,0,82725
compass=0,3,2,4,5,1,8,7,6,0,45717
compass=0,7,6,8,1,4,5,2,3,0,45890
compass=0,6,8,1,7,5,4,2,3,0,81148
compass=0,7,6,8,1,2,3,4,5,0,90405
compass=0,3,4,5,7,6,8,1,2,0,67261
compass=0,1,2,3,8,4,5,6,7,0,51524
compass=0,1,2,3,5,4,7,6,8,0,67333
compass=0,1,2,3,4,5,7,6,8,0,58432
compass=0,6,7,3,2,5,4,8,1,0,91771
compass=0,1,3,8,2,5,4,6,7,0,2859
compass=0,1,3,8,2,4,5,6,7,0,14051
compass=0,5,4,7,6,8,1,2,3,0,10597
compass=0,1,2,3,4,5,7,6,8,0,38021
compass=0,7,6,5,4,3,2,1,8,0,73122
compass=0,4,3,2,1,8,7,6,5,0,92658
compass=0,5,2,6,7,4,1,3,8,0,59237
compass=0,8,5,6,4,7,2,1,3,0,28309
compass=0,8,5,6,4,7,2,1,3,0,68821
compass=0,6,7,4,5,1,2,3,8,0,5130
compass=0,7,6,5,8,1,3,2,4,0,58993
compass=0,1,8,2,3,5,4,7,6,0,42834
compass=0,1,2,3,4,5,7,6,8,0,65110
compass=0,6,8,4,5,1,2,3,7,0,90723
compass=0,6,8,4,5,1,2,3,7,0,27309
compass=0,5,7,6,8,1,2,3,4,0,37321
compass=0,8,7,6,5,4,3,2,1,0,76679
compass=0,6,5,4,3,2,1,8,7,0,51287
compass=0,3,2,1,8,6,7,5,4,0,39774
compass=0,6,5,4,8,1,2,3,7,0,43124
compass=0,3,4,5,7,6,8,1,2,0,57109
compass=0,2,6,1,7,3,5,4,8,0,30311
compass=0,6,7,3,2,1,4,5,8,0,54699
compass=0,6,3,8,5,1,4,2,7,0,46504
compass=0,1,2,3,4,5,7,6,8,0,70832
compass=0,3,4,5,7,6,8,1,2,0,75108
compass=0,8,3,2,1,5,6,4,7,0,83118
compass=0,1,3,8,4,5,2,6,7,0,61265
compass=0,3,2,4,7,5,1,6,8,1,59741
compass=0,3,2,6,7,5,4,1,8,0,51609
compass=0,1,2,3,4,5,6,7,8,0,21408
compass=0,1,2,3,4,5,7,6,8,0,62853
compass=0,1,2,3,7,4,5,6,8,0,62714
compass=0,1,2,3,4,5,8,6,7,0,39427
compass=0,6,7,8,1,2,3,4,5,0,17915
compass=0,1,2,3,4,5,7,6,8,0,85609
compass=0,1,2,3,4,5,7,6,8,0,87967
compass=0,1,2,3,4,5,6,7,8,0,3890
compass=0,3,2,7,8,4,1,6,5,0,14278
compass=0,1,2,3,6,7,5,4,8,0,31993
compass=0,8,1,2,3,6,7,5,4,0,14487
compass=0,1,3,2,5,4,7,8,6,0,24052
compass=0,1,2,3,4,5,6,7,8,0,66013
compass=0,5,6,7,8,1,2,3,4,0,92325
compass=0,7,6,8,1,2,3,5,4,0,27833
compass=0,6,1,5,3,4,8,2,7,1,51397
compass=0,4,5,8,6,7,1,2,3,0,9143
compass=0,6,7,5,4,3,2,1,8,0,42313
compass=0,3,4,7,5,6,8,1,2,0,27014
compass=0,6,1,5,2,8,4,3,7,0,33279
compass=0,6,8,4,5,2,1,3,7,0,60699
compass=0,6,7,4,3,1,2,5,8,0,41877
compass=0,1,2,6,4,3,5,8,7,0,57284

Appendix F. HSV Flowchart



Appendix G. CIMC Flowchart



Appendix H. CIMC Code

```
%=====
%
% Name:  CIMC - Change In Mean Center
%
% Type:  MATLAB Function
%
% Author:  Keith French
%
% Synopsis:  This function will generate an RGB color composite matrix of the
%            same dimensions as the input matrices based on the change in
%            the weighted mean center in moving windows from one matrix to
%            the other. The direction component of the change vector is
%            symbolized by the hue, the magnitude component is symbolized by
%            the value.
%
% Syntax:  [rgbCompositeOutput] = cimc(densGrid1, densGrid2, nbhdSize, linLog);
%
% Inputs:  densGrid1 - Density grid 1
%          densGrid2 - Density grid 2 (should be the same dimensions as grid 1)
%          nbhdSize - Moving window size (a positive integer)
%          linLog - 0 applies a linear scaling on the magnitude (value) component
%                 1 applies a logarithmic scaling on the magnitude component
%
% Output:  An [inX, inY, 3] dimensional matrix as an RGB composite
%
%=====

function [rgbCompositeOutput] = cimc(densGrid1, densGrid2, nbhdSize, linLog);

    % Calculate focal mean centers of population in moving windows
    [x1, y1] = focalpopcent(densGrid1, nbhdSize);
    [x2, y2] = focalpopcent(densGrid2, nbhdSize);

    % Calculate direction and magnitude grids
    dir = direction(x1, y1, x2, y2);
    mag = magdiff(x1, y1, x2, y2);

    % Convert direction grid to hue grid
    h = dir2hue(dir);

    % Generate constant saturation grid
    s = ones(size(h));

    % Convert magnitude grid to hue grid
    v = mag2val(mag, linLog);

    % Convert hsv to rgb
    [r, g, b] = hsv2rgb(h, s, v);

    % Output final composite
    rgbCompositeOutput = cat(3, r, g, b);

end

%=====
%
% L O C A L    F U N C T I O N S
%
%=====

function [x, y] = popcent(popgrid)

    % USAGE: [x, y] = popcent(popgrid) where popgrid is a matrix with population
    % density values (or any value you wish to act as a weight) and [x, y] are
    % the x and y values of the resultant population center relative to the
    % input grid with the origin at the upper left.

    rows = size(popgrid, 1);
```

```

cols = size(popgrid, 2);

wsum = sum(popgrid(:));

xsum = 0; ysum = 0;
for i = 1:rows
    for j = 1:cols
        xsum = xsum + popgrid(i,j).*j;
        ysum = ysum + popgrid(i,j).*i;
    end
end

x = xsum / wsum;
y = ysum / wsum;

end

%-----

function [xgrid, ygrid] = focalpopcent(popgrid, window_size)

% USAGE [xgrid, ygrid] focalpopcent(popgrid, window_size) where popgrid is
% the input grid and window_size is an positive integer representing the size of
% the focal region. Code will produce warnings if window_size is an even
% integer but will still run. Outputs two grids, one for the x location and one
% for the y location.

rows = size(popgrid, 1);
cols = size(popgrid, 2);

buffer = (window_size - 1) / 2;
for i = 1:rows
    for j = 1:cols

        % determine neighborhood region
        xstart = j - buffer;
        if (xstart < 1)
            xstart = 1;
        end
        xend = j + buffer;
        if (xend > cols)
            xend = cols;
        end
        ystart = i - buffer;
        if (ystart < 1)
            ystart = 1;
        end
        yend = i + buffer;
        if (yend > rows)
            yend = rows;
        end
        nbhood = popgrid(ystart:yend, xstart:xend);
        [x, y] = popcent(nbhood);

        xgrid(i,j) = xstart - 1 + x;
        ygrid(i,j) = ystart - 1 + y;

    end
end

end

%-----

function [dir] = direction(xgrid1, ygrid1, xgrid2, ygrid2)

% USAGE [dir] = direction(xgrid1, ygrid1, xgrid2, ygrid2) where the input
% grids consist of x and y coordinates of two different snapshots (1 and 2).
% The output is a single grid indicating the directional changes between
% snapshots in radians (-pi to pi range).

```

```

    x = xgrid2 - xgrid1;
    y = ygrid1 - ygrid2;

    dir = atan2(y, x);

end

%-----

function [mag] = magdiff(xgrid1, ygrid1, xgrid2, ygrid2)

    % USAGE [mag] = magdif(xgrid1, ygrid1, xgrid2, ygrid2) where the input
    % grids consist of x and y coordinates of two different snapshots (1 and 2).
    % The output is a single grid indicating the magnitude changes between
    % snapshots in positive floats.

    mag = sqrt((xgrid1 - xgrid2).^2 + (ygrid1 - ygrid2).^2);

end

%-----

function [huegrid] = dir2hue(dirgrid)

    % USAGE [huegrid] = dir2hue(dirgrid) where dirgrid is a double float grid
    % with a range of values from -pi to pi. The output is a double float grid
    % with values from 0 to 1.

    hg = 0.75 - ((dirgrid + pi) ./ (2*pi));
    huegrid = hg;
    huegrid(hg<0) = hg(hg<0) + 1;

end

%-----

function [valgrid] = mag2val(maggrid, linlog)

    % USAGE [valgrid] = mag2val(maggrid, linlog) where maggrid is a double float
    % grid with positive double float values and linlog is 0 for a linear scaling
    % of magnitude or 1 for a logarithmic scaling of magnitude. The output (valgrid)
    % is double float grid with values from 0 to 1.

    if (linlog)
        grid = log(maggrid + 1);
    else
        grid = maggrid;
    end

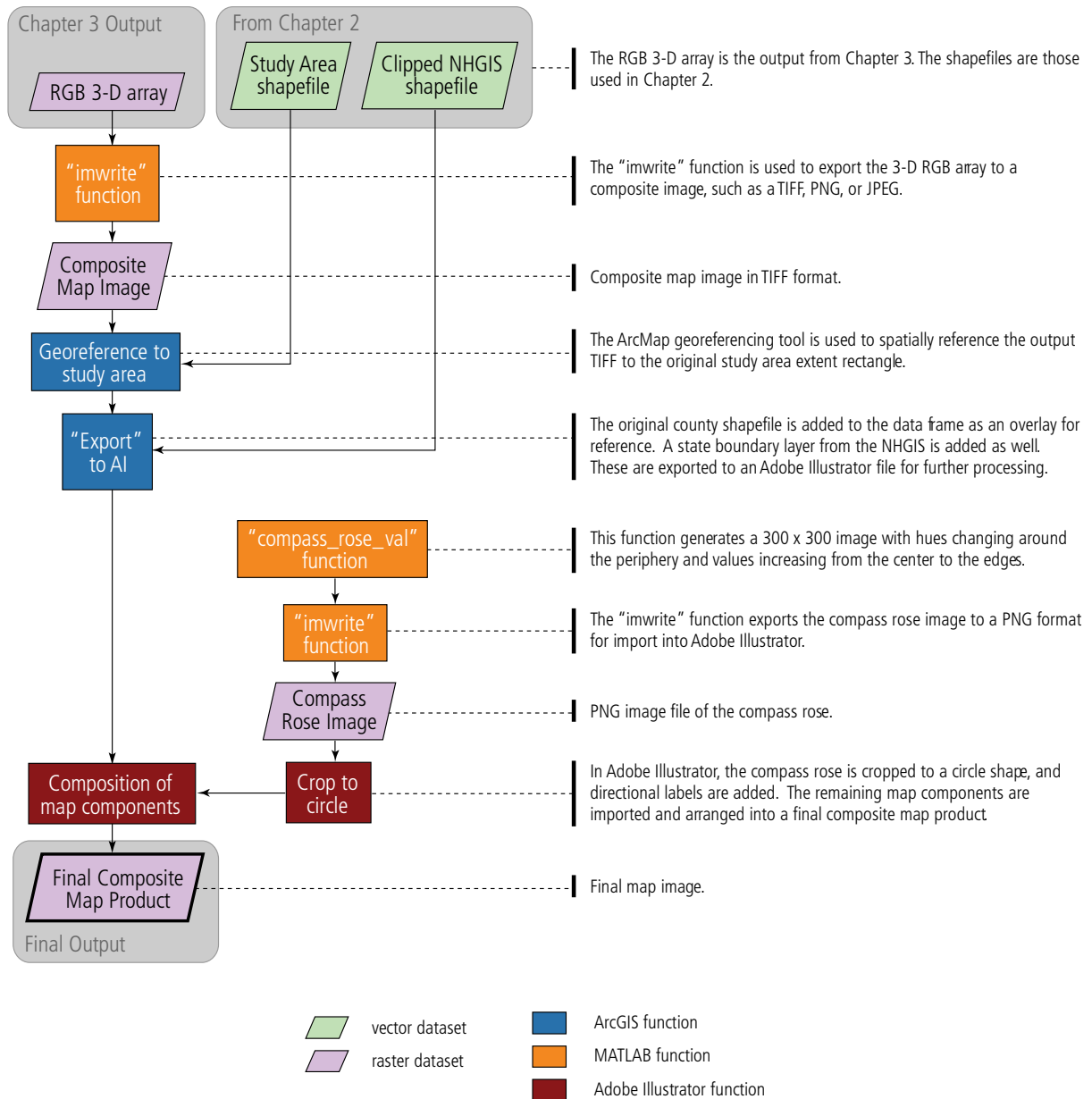
    maxpos = max(grid(:));
    minpos = min(grid(:));
    rngpos = maxpos - minpos;

    valgrid = ((grid - minpos) ./ rngpos);

end

```

Appendix I. Post-processing Flowchart



Appendix J. The Map Survey

<----- IMPORTANT: PLEASE MAXIMIZE OR RESIZE THIS WINDOW UNTIL ALL OF THIS RED TEXT IS ON ONE LINE ----->

The following survey is an assessment of a mapping technique developed as part of a master's thesis. Please review the map figures presented, carefully read the legends to the right of each map, and answer the questions to the best of your ability. There are eight questions, each referring to the yellow lettered circles in maps, and you will need to refer back to the figures and the accompanying legends (to the right of each map) to answer the questions. Please, DO NOT TAKE THE SURVEY MORE THAN ONE TIME, but feel free to forward the link to colleagues or friends whom you think may be interested in helping out with this.

Information Statement:

The Department of Geography at the University of Kansas supports the practice of protection for human subjects participating in research. The following information is provided for you to decide whether you wish to participate in the present study. You should be aware that even if you agree to participate, you are free to withdraw at any time without penalty.

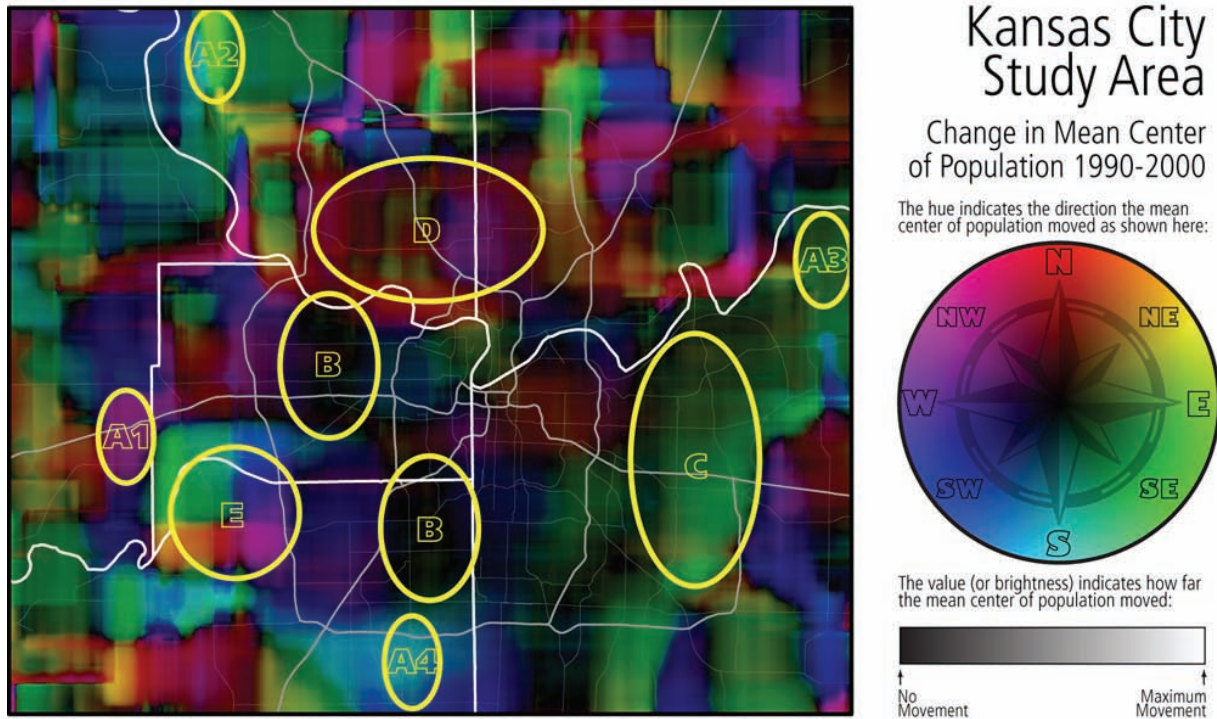
We are conducting this study to better understand the legibility and interpretation of a new method of population change mapping. This will entail your completion of a short survey. Your participation is expected to take approximately 10-15 minutes to complete. The content of the survey should cause no more discomfort than you would experience in your everyday life.

Although participation may not benefit you directly, we believe that the information obtained from this study will help us gain a better understanding of communication through cartographic methods. Your participation is solicited, although strictly voluntary. Your name will not be associated in any way with the research findings. Your identifiable information will not be shared unless (a) it is required by law or university policy, or (b) you give written permission. Please read the Survey Monkey privacy statement for more information.

If you would like additional information concerning this study before or after it is completed, please feel free to contact us by e-mail: Principal investigator: Keith French (kaf@ku.edu), faculty supervisor: Dr. Xingong Li (lixixi@ku.edu).

Completion of the survey indicates your willingness to take part in this study and that you are at least 18 years old. If you have any additional questions about your rights as a research participant, you may call (785) 864-7429 or write the Human Subjects Committee Lawrence Campus (HSCL), University of Kansas, 2385 Irving Hill Road, Lawrence, Kansas 66045-7563, email irb@ku.edu.

Please click Next to begin.



Map 1: Kansas City Study Area

1. Which of the following areas appear to have the highest magnitude of mean center of population movement?

- a) A1
- b) A2
- c) A3
- d) A4

2. Which answer best describes what the map depicts in the two areas marked "B"?

- a) The population increased
- b) The population stayed the same
- c) The mean centers of population moved very little
- d) There are no inhabitants in these locations

3. Which of the four following directions best describes the movement of mean centers of population occurring in area "C"?

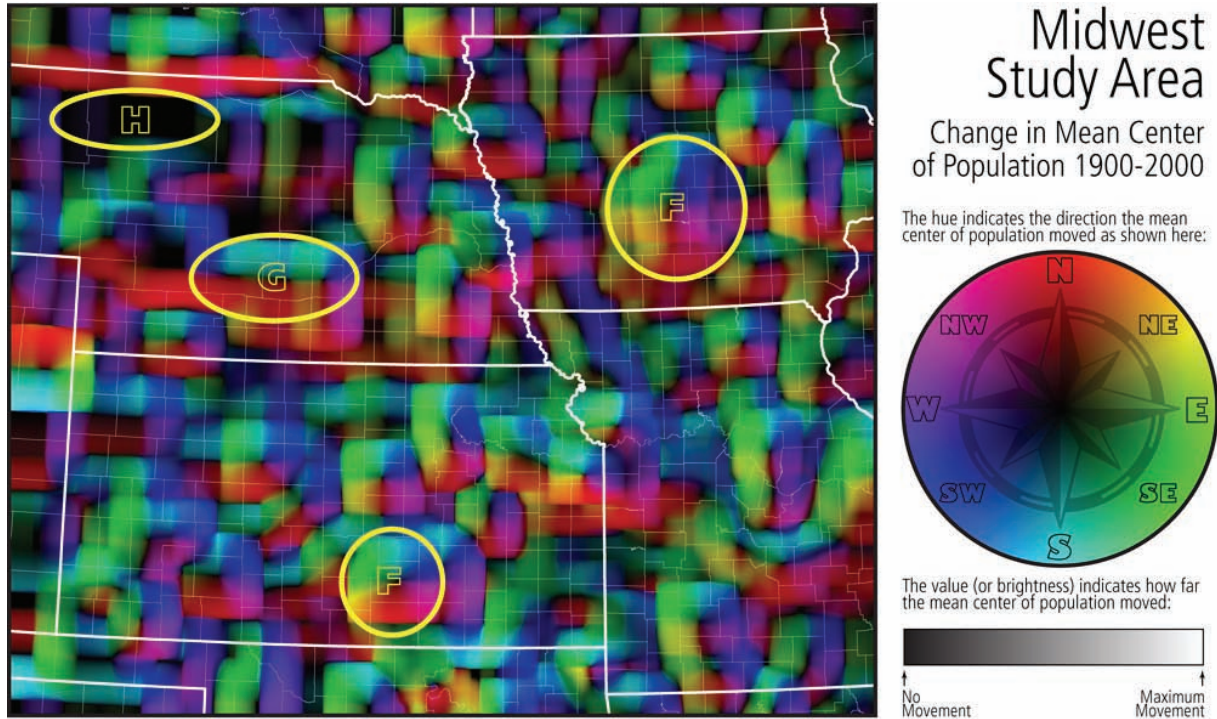
- a) Northeast (NE)
- b) Southeast (SE)
- c) South (S)
- d) No direction is indicated

4. Based on what the map is symbolizing, in general, what is happening in area "D"?

- a) The population is growing
- b) The population is shrinking
- c) The mean centers of population have moved in a mostly easterly direction
- d) The mean centers of population have moved in a mostly northerly direction

5. Which answer best describes what the map depicts in area "E"?

- a) There was a sharp drop in population
- b) The mean centers of population shifted inward toward the center of area "E"
- c) The mean centers of population shifted outward away from the center of area "E"
- d) The mean centers of population moved very little



Map 2: Midwest Study Area

6. In general, from 1900 to 2000, Midwest rural populations grew little or declined, while urban populations exploded in growth. Based on this knowledge and the information in the map, what are most likely centered on the two areas marked "F"?

- a) Abandoned areas
- b) Urban areas
- c) Rural areas
- d) None of the above

7. Interstate 80 runs east-west through the center of area "G", and was completed in the early 1970s. What is the most likely scenario based on what is shown in the map?

- a) The construction of the interstate had little influence on the population
- b) The construction of the interstate caused people to move away from it
- c) The construction of the interstate caused people to move toward it
- d) None of the above

8. In the northwest corner of the map is an area marked "H". Please briefly describe, in your own words, what the map is depicting in this area.

Appendix K. The Map Survey Results

Respondent	Score	P	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	TRY
0	8	B	C	B	D	B	B	C		Little or no change in the mean center of population.	N/A
3	8	N	B	C	B	D	B	B	C	little or no change in mean center of population	Yes
8	8	N	B	C	B	D	B	B	C	Static population	Yes
13	8	N	B	C	B	D	B	B	C	There has been no movement in this area.	Yes
15	8	N	B	C	B	D	B	B	C	An area that was and still is sparsely populated with little movement occurring over the last 100 years.	Yes
17	8	N	B	C	B	D	B	B	C	Either no movement in any direction, or no data	Yes
28	8	Y	B	C	B	D	B	B	C	No movement.	Yes
33	8	Y	B	C	B	D	B	B	C	people stayed where they are, or the area is/was never populated	Yes
38	8	Y	B	C	B	D	B	B	C	There is very little to no movement of the mean center of population.	Yes
40	8	Y	B	C	B	D	B	B	C	The center of population has not moved.	Yes
44	8	Y	B	C	B	D	B	B	C	No movement of the mean population	Yes
50	8	N	B	C	B	D	B	B	C	No population movement	Yes
1	7	N	D	C	B	D	B	B	C	You are seeing little if any movement of population center in the Sand Hills. Low population there anyway.	Yes
7	7	N	B	C	A	D	B	B	C	little to no population growth or migration	Yes
11	7	N	B	C	A	D	C	B	C	There is very little to no movement to this area	Yes
41	7	Y	B	C	B	D	C	B	C	no movement	Yes
42	7	Y	B	C	B	C	B	B	C	MCP staying constant or very little movement.	Yes
4	6	N	D	C	B	D	B	C	C	No movement in this area.	Yes
12	6	N	C	C	B	D	B	C	C	Population mainly stayed the same	Yes
14	6	N	B	C	B	B	B	B	A	Very little movement	Yes
25	6	Y	D	C	B	D	A	B	C	There was very little movement in that area	Yes
26	6	Y	B	C	B	D	C	B	C	It is depicting a rural area	Yes
27	6	Y	B	C	B	A	C	B	C	No movement	Yes
36	6	Y	B	C	B	D	D	B	C	The area is basically unpopulated	Yes
16	5	N	C	C	B	C	B	C	C	there wasn't much movement	Yes
29	5	Y	D	C	B	C	C	B	C	No movement	Yes
32	5	Y	D	C	B	D	C	C	C	almost no movement of the population	Yes
37	5	Y	B	C	C	B	D	B	C	little to no movement in population	Yes
45	5	Y	B	C	B	D	A	C	D	It is an area with no movement	Yes
46	5	Y	B	B	B	B	B	D	C	Either uninhabital or no movement at all	Yes
48	5	Y	B	C	B	D	D	D	D	it's dark so not much movement there	Yes
2	4	N	B	B	B	D	B	D	A		No
5	4	N	A	C	B	D	C	C	D	No movement	Yes
10	4	N	B	B	B	D	C	C	B	No movement	Yes
34	4	Y	B	C	C	D	C	B	B	N/A	No
35	4	Y	B	B	B	B	C	B	B	little to no movement	Yes
39	4	Y	B	C	B	D	C				No
9	3	N	B	B	B	C	A	D	B	No movement	Yes
18	3	N	C	A	C	A	D	B	C	Very little population change	Yes
24	3	Y	C	B	B	B	C	B	B	No movement	Yes
30	3	Y	B	A	C	C	B	B	A	population loss	Yes
31	3	Y	B	B	B	A	D	D	D	no movement in the more populations.	Yes
6	2	N	B	B	B	B	C				No
20	2	Y	B	D	B	C	D	D	D	idk	No
23	2	Y	A	A	A	D	B	C	B	Rural areas that people are moving away from	Yes
43	2	Y	A	B	D	D	B	A	B	Population moved north and west, congregating more in this locale.	Yes
21	1	Y	C	B	C	A	C	B	A	none	No
22	1	Y	A	B	B	A	A	D	D	Unknown	No
47	1	Y	C	A	B	C	C				No
19	0	Y	A	A	A	A	A	A	A	UFO's	No
49	0	Y	A	A	A	A	A	A	A	yes	No

Respondent: This number indicates the order in which the respondent took the survey. The number 0 was not an actual respondent; the values in this row are the correct answers to the questions.

Score: The number of questions answered correctly (highlighted in green) out of 8 possible.

P: This indicates whether or not the respondent was part of the paid "targeted audience". This was made evident by the SurveyMonkey individual result data.

Q1-Q8: The respondents' answers to questions 1-8

TRY: This indicates whether or not the respondent was deemed to be taking the survey seriously.